CR-151328

ENGINEERING STUDY OF THE DIGITAL-OPERATIONAL TECHNIQUE

FINAL REPORT

15 September 1964

Submitted to:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MANNED SPACECRAFT CENTER

STD, CONTROL AND COMPUTER SYSTEMS

HOUSTON, TEXAS

CONTRACT NAS 9-2522

(NASA-CR-151328) ENGINEERING STUDY OF THE DIGITAL-OPERATIONAL TECHNIQUE Final Report (Laboratory for Electronics, Inc.) 152 p

N77-76881

Unclas
00/04 24462

LFE ELECTRONICS

A DIVISION OF LABORATORY FOR ELECTRONICS, INC.

BOSTON 15, MASSACHUSETTS

C-471

ENGINEERING STUDY OF THE DIGITAL-OPERATIONAL TECHNIQUE

FINAL REPORT

15 September 1964

Submitted to:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MANNED SPACECRAFT CENTER

STD, CONTROL AND COMPUTER SYSTEMS

HOUSTON, TEXAS

CONTRACT NAS 9-2522

LFE ELECTRONICS

A Division of LABORATORY FOR ELECTRONICS, INC.

Boston, Massachusetts 02215

ACKNOWLEDGEMENTS

This program was monitored by Mr. C. Brady of NASA, MSC, Houston, Texas, whose aid during the course of this program was most helpful and appreciated.

Special mention of the following project co-workers is warranted for their many contributions to the success of the program.

P. James implemented the IBM 1401 Runs and assisted in the result analysis. F. Smith was responsible for implementing the MEDOC programs and supervising the maintenance of the machine. E. Schnall was the major contributor for the Estimate of an Airborne Computer, Task III; D. Prosser and T. Finnegan also contributed to this task. C. Battarel provided the major contribution for the Error Analysis, Task I. M. Elmer assumed program-management responsibilities in addition to major contributions for the MEDOC Simulation, Task II.

The authors are grateful for the cooperation and contributions of all project personnel.

FOREWORD

This research program, initiated by STD, Control and Computer Systems of NASA's Manned Spacecraft Center, under Contract No. NAS 9-2522, called for LFE Electronics to study the Digital-Operational Computer Technique in terms of: an error analysis, simulation of a special-purpose computer problem, and an estimate of the weight, size, power and component count characteristics of an airborne computer.

Due to NASA's interest in strapped-down guidance systems, the direction-cosine, coordinate-conversion computer problem was selected to be studied by LFE. Strap-down computation may be divided into two parts – attitude and navigation. The navigation computation keeps track of vehicle position in inertial coordinates and may be performed at a relatively slow rate. The attitude computation, however, relates the rapidly rotating vehicle axis to the fixed, inertial-coordinate system and, therefore, must be performed at high speed. It is the attitude computation that is unique to a strap-down configuration and upon which over-all system accuracy largely depends; thus, attitude computation is pertinent to this study and the following discussion.

This report shall not delve into the question: "Should a general-purpose computer be used to solve the attitude equations?" It does, however, become involved with the question "Which technique?", which arises once a decision favorable to a special-purpose computer has been made. Hopefully, this report will assist the reader in answering the latter question.

Presented herein are the results of a 5-month study of the Digital-Operational (DO) computation technique. For those readers who are not familiar with the DO technique, it is suggested that References 1, 2 and 6 in the Bibliography (Appendix D) be studied prior to reading this report. The following sections are logically arranged to afford the reader a ready understanding of the program's

scope without overconcern for its mathematical detail. Those readers who are interested in details and back-up justification of the comments made, are referred to the Appendices. The report describes the three tasks to be performed, the results obtained, conclusions, and, finally, recommendations for future efforts.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	ii
FOREWORD	iii
1.0 INTRODUCTION	1-1
2.0 ERROR PREDICTION, TASK I	2-1
2.1 Method of Approach	2-1
2.2 Error Sources	2-2
2.2.1 Truncation Error	2-2
2.2.2 Field Error	2-2
2.2.3 Integrand Biased Quantization Error	2-2
2.2.4 Integrand Random Quantization Error	2-3
2.2.5 Integrand Sign-Reversal Error	2-3
2.2.6 Total Error Sources	2-3
2.3 Error Propagation	2-4
2.3.1 Error Propagation Analytical Model	2-4
2.3.2 Error Prediction for the Free-Running Sine-Cosine Generator	2-6
2.3.3 Error Prediction for the Alternating (Limit Cycles) Sine-Cosine Generator	2-7
2.4 Direction Cosines – Error Prediction In the General Case of a 3-Angle Variation	2-8
2.4.1 Error Sources	2-8
2.4.2 Error Propagation	2-10
2.4.3 Error Prediction for Limit Cycles of the Direction-Cosine Generator	2-12
2.5 Comparison of Error Prediction and Experimental Data	2-13
2.6 Reversible Computation for Small Limit Cycles	2-17
3.0 MEDOC SIMULATION, TASK II	3-1
3.1 Statement of the Problem	3-1

TABLE OF CONTENTS (Cont.)

	Page
3.2 MEDOC Program	3-5
3.3 Scaling and Smoothing	3-7
3.4 Initial Conditions	3-10
4.0 ESTIMATE OF AN AIRBORNE COMPUTER, TASK III	4-1
4.1 Basis of Estimates	4-1
4.2 Summary of Results	4-1
4.3 Interpretation of Results	4-8
5.0 CONCLUSIONS	5-1
6.0 RECOMMENDATIONS	6-1
6.1 Study Areas	6-1
6.2 Hardware Areas	6-1
APPENDIX A:	
A.1 DO PULSE MULTIPLIER ANALYSIS: CASE OF CONSTANT Y-COUNTER	A-1
A.2 DO WORST-CASE ERROR FORMULAE: CASE OF VARYING Y-COUNTER	A-7
A.3 DIGITAL-OPERATIONAL QUASI-WORST-CASE ERROR FORMULAE FOR OPEN-LOOP INTEGRATION	A-14
A.4 DO CLOSED-LOOP PROPAGATED ERRORS	A-28
APPENDIX B. ERROR CURVES; MEDOC PROGRAM RUNS	B-1
APPENDIX C. AIRBORNE COMPUTER ANALYSIS	C-1
APPENDIX D. BIBLIOGRAPHY	D-1

LIST OF ILLUSTRATIONS

Figure		Page
3.1	MEDOC (Modular-Expandable, Digital-Operational Computer) Breadboard	3-2
3.2	Strap-Down System Attitude Computation	3-3
3.3	MEDOC Program	3-6
3.4	Pulse Multiplier Timing Diagram	3-9
3.5	Pulse Multiplier Output with Digital Smoothing	3-11
3.6	Sine-Cosine Error Curve – Smoothing vs. Non-Smoothing	3-12
4.1	Clock Performance Profile	4-2
4.2	Component Count	4-4
4.3	Power Drain	4-5
4.4	Weight	4-6
4.5	Volume	4-7
5.1	Expected and Worst-Case Errors, Predicted and Experimental (Rotation)	5-2
5.2	Expected and Worst-Case Errors, Predicted and Experimental (Limit Cycle)	5-3
5.3	Expected and Worst-Case Errors, Predicted and Experimental (Limit Cycle $\Delta\theta_i = 2^{-15}$)	5-4
A.1	DO Pulse Multiplier Logic Diagram	A-2
A.2	DO Square Generator	A-11
A.3	DO Integrator Error Terms	A-17
A.4	Scale Factors	A-19
A.5	Form Factors	A-20
A.6	Coupled Closed Iteration Loops	A-32
B.1	Program Run 1.101	B-4
B.2	Program Runs 1.109 & 1.201	B-5
B.3	Program Runs 2.101 & 2.102	B-9
B.4	Program Run 3.101	B-13
B.5	Program Runs 3.102 & 3.108	B-14
B.6	Program Run 3.208	B-15

LIST OF ILLUSTRATIONS (Cont.)

Figure		Page
B.7	Program Runs 3.103, 3.104, 3.106, 3.107	B-16
B.8	Program Runs 3.202 & 3.203	B-17
B.9	Direction-Cosine Vector Diagram	B-19
C.1	Block Diagram of Counter FBC-8D for T-1 Computer	C-2
C.2	Block Diagram of Sub-Multiplier MS-15 for T-1 Computer	C-2
C.3	Block Diagram of Product Adder MP/A for T-1 Computer	C-3
C.4	Block Diagram of Synchronizer SI-3; Each of Three Channels	C-4
C.5	Block Diagram of Upscaler	C-4
C.6	Block Diagram of Clock Slot Generator for T-1 and T-2 Computers	C-5
C.7	Block Diagram of Counter FBC-6 for T-2 Computer	C-5
C.8	Block Diagram of Counter Stages for F-1 Computer	C-8
C.9	Block Diagram of Sub-Multiplier MS-19 for F-1 Computer	C-9
C.10	Block Diagram of Product Adder MP/A for F-1 Computer	C-10
C.11	Block Diagram of Counter Stages for F-2 Computer	C-11
C.12	Block Diagram of Sub-Multiplier MS-20 for F-2 Computer	C-12
C.13	Block Diagram of Product Adder MP/A for F-2 Computer	C-13
C.14	Power Supply Block Diagram	C-15
C.15	Typical Computer Assembly	C-18
C.16	Microelectronic Subassembly Repair	C-22

1.0 INTRODUCTION

This report presents the results of a study of the application of the DO technique to the attitude computation problem associated with a strap-down guidance system.

The DO technique is by no means a new innovation, and, as noted in the Bibliography, has been investigated by many others. This is a technique that, by its ability to operate in either the rate or incremental modes, is competitive with the DDA and other special-purpose incremental computers. For the same accuracy and time of solution, the DO technique should demonstrate a 25 percent potential reduction in size, weight, and power requirements over the DDA implementation. This results from the fact that where the DDA requires two registers per integrator, the DO technique allows sharing the independent variable register with several registers in the rest of the program.

The study was divided into three tasks:

TASK I, ERROR ANALYSIS – is concerned with developing error formulae in the general case which can be used to predict the errors for any DO mapping configuration.

TASK II, MEDOC SIMULATION – concerns the programming of the direction cosine and coordinate transformation problem on the Modular-Expandable, Digital-Operational Computer which is located at LFE. Program sets were written which were applicable to machine runs on both the IBM 1401 and MEDOC. The MEDOC results were compared with the reference IBM results and the error curves were thus generated. The predicted errors based upon the error formulae, as derived in Task I, were compared to the machine runs.

TASK III, ESTIMATE OF AN AIRBORNE COMPUTER – is the study of the implementation of the attitude equations utilizing the Texas Instruments flat packs and the Fairchild Micrologic units. Estimates of volume, weight, power, and component count are included. Conclusions and recommendations for further work are also included in this report.

After reviewing this report, the reader should be in a position to compare the results of this DO study with that of a DDA or any other special-purpose computer technique for the attitude computation application. In addition, the error formulae, as derived in Task I, are applicable to any DO mapping configuration.

2.0. ERROR PREDICTION, TASK I

2.1. Method of Approach

An exact error prediction requires a pulse-by-pulse analysis of the digital-operational (DO) problem implementation. This analysis, although possible as shown in Appendix A.1, would be highly impractical. The difficulty stems from the very functioning of the pulse multiplier where the output is an irregular function of the inputs. The use of an exact error formula for a given case – for example, the sine-cosine generator with given scale factors and initial values – will not give any indication of the error obtained for other initial conditions. To obtain a more generally applicable error formulae, a more statistical point of view must be introduced. An order of magnitude of the errors will be obtained with boundaries on the upper values, according to the analysis developed in Appendices A.2 and A.3. Error source formulae and open-loop propagation formulae are derived in Appendix A.3. When the iteration path is closed, as is the case for the direction-cosine problem, the formulae must be modified in order to take into account the error-feedback action. Closed-loop error prediction formulae are derived in Appendix A.4. However, these formulae apply to the case where the error build-up through error propagation is larger than the error sources themselves. This is not the case of the sine-cosine generator where there is negative feedback and the error relaxation (defined in Appendix A.4, para. A.4.7.4) interval is of 2 radians with a 1-pulse amplitude, and the error sources generate several error pulses within 2 radians. In this conjecture, a specialized error-propagation formula must be derived that will be applicable only to the studied function – the direction-cosine generator. In the following, the error sources will be analyzed, then a specializer error-propagation formula will be derived for the sine-cosine generator, and, finally, it will be extended to the general case of the direction-cosine generator where the three angles driven will be discussed.

2.2. Error Sources

The error sources correspond to the open-loop error formulae developed in Appendices A.2 and A.3. They will first be applied to the case of a sine-cosine generator for a computation time of 20 revolutions ($\theta = 40\pi$ radians) with 17-bit counters and 17-bit multipliers.

2.2.1. Truncation Error. The sine-cosine generator is a DO program where the rate of the dx input is always larger then the rate of the dy input. Therefore, the worst case of an equal rate, described in Appendix A.2 is avoided and the worst-case truncation-error problem will be given by Eq. (A-15).

$$\xi = \sqrt{\sum_{ij=1}^{k} (2^{c} DY_{ij}\, m)^2} \quad .$$

For the free-running sine-cosine generator, the number of ij intervals correspond to the number of monotonic variations without sign reversal of the integrand, and for sin θ as well as cos θ. For 20 revolutions, $k = 4 \times 20 = 80$. Observing that a scale factor of 0.9 can be described by a scale factor of one and a Y_{max} of 0.9 instead of one, Eq. (A-15) becomes

$$\xi = \sqrt{(.9)^2 \times (17)^2 \times 80} = 137 \text{ pulses} \qquad (2\text{-}1)$$

2.2.2. Field Error. There is no field error in this problem, since the entire field of the sin θ and cos θ counters is used for multiplier inputs.

2.2.3. Integrand Biased Quantization Error. When the Y-counter scale factor is chosen to be 2^{-1}, the initial values of sin θ and cos θ can be set exactly and $\zeta = 0$, for $\theta = 0$. However, in the case of a scale factor, 0.9, or when the initial values cannot be described by a power of two, the initial values are set with an accuracy of ± 1/2 pulse. In open-loop integration, a large error could result such as $\zeta = 1/2 \times 40\pi = 63$ pulses. Since the problem involves closed-loop integration with negative feedback (cf. Appendix A.4) and a relaxation time

t_r = 2 radians, this initial error will be computed after 2 radians and will not grow any further. For all practical purposes, this error can be neglected.

2.2.4. Integrand Random Quantization Error. Application of Eq. (A-21), developed in Appendix A.3 gives

$$\epsilon = \frac{1}{\sqrt{12}} \quad \frac{\Sigma |w_j|}{\sqrt{\Sigma |v_i|}} \times 2^{-\frac{m}{2} + p - \frac{q}{2} - a + \frac{b}{c} - \frac{c}{2}} \tag{2-2}$$

$$\epsilon = \frac{40\pi}{3.46(6)} \simeq 6 \text{ pulses} \quad .$$

2.2.5. Integrand Sign-Reversal Error. In the worst-case rule of Appendix A.2, one output error pulse is accounted for in each integrand sign reversal. In the sine-cosine generator, however, it is very unlikely that this worst case will be met, since, due to errors, the maximum values do not repeat exactly. Therefore, a statistical reduction will be obtained and the estimated average integrand sign-reversal error will be

$$e_{SR} = \sqrt{SR} = \sqrt{40} = 6.3 \text{ pulses} \quad . \tag{2-3}$$

2.2.6. Total Error Sources. The sign of the errors are unknown; only their magnitude predicted at the end of the 20th revolution, is known to be:

$$e_T = \xi + \epsilon + e_{SR} = 150 \text{ pulses} \tag{2-4}$$

for the sin θ counter as well as the cos θ counter. Such errors are large and are regenerated within each monotonic interval with different signs. It is therefore not clear as to what will be the action of negative feedback, as analyzed in Appendix A.4. Further insight into the actual form of the error function will be obtained through use of a specialized error-propagation formula in the next section.

It can be seen that use of smoothing, where the last two stages of the sin θ and cos θ counters are not read out, reduces the total error source to the 150/4 value.

$$\bar{e}_T \text{ (smoothing)} = 38 \text{ pulses} \quad . \tag{2-5}$$

2.3. Error Propagation

2.3.1. Error Propagation Analytical Model. An inspection of the program map (Figure 3.3) permits the equations for the iterated values of sin θ and cos θ for intervals of computation of several input pulses, dθ, to be written. Let θ_1 and θ_2 be two successive positions of θ, e_{s_1} and e_{s_2} the corresponding errors on the values of sin θ_1 and sin θ_2, and e_{c_1} and e_{c_2} the errors on the values of cos θ_1 and cos θ_2. The corresponding total error source in multiplier M_S feeding the sin θ counter being $\xi_s^{1\to 2}$. It can be written by inspection

$$\sin\theta_2^* = \sin\theta_1 + e_{s1} + \int_{\theta_1}^{\theta_2} [\cos\theta_1 + e_{c_1} - \int_{\theta_1}^{\theta} (\sin\theta + e_{s_1})\, d\theta - \xi_c^{1\to 2}]\, d\theta + \xi_s^{1\to 2} \tag{2-6}$$

$$\cos\theta_2^* = \cos\theta_1 + e_{c_1} - \int_{\theta_1}^{\theta_2} [\sin\theta_1 + e_{s_1} + \int_{\theta_1}^{\theta} (\cos\theta + e_{c_1})\, d\theta + \xi_s^{1\to 2}]\, d\theta - \xi_c^{1\to 2} \tag{2-7}$$

where sin θ^*_2 and cos θ_2^* are the obtained functions, while sin θ_2 and cos θ_2 are the desired functions.

Equations (2-6) and (2-7) can be solved only if the functions $\xi_s^{1\to 2}$ and $\xi_c^{1\to 2}$ are known. However, it has been shown in Section 2.2 that only the average magnitude of the ξ's are known. The analysis must therefore be restricted to

the study of the propagated error, independent of the error sources. In this case, Equations (2-6) and (2-7) become

$$\sin \theta^*_2 = \sin \theta_2 + e_{s_1} + e_{c_1} (\theta_2 - \theta_1) - \frac{e_{s_1} (\theta_2 - \theta_1)^2}{2} \qquad (2\text{-}8)$$

$$\cos \theta^*_2 = \cos \theta_2 + e_{c_1} - e_{s_1} (\theta_2 - \theta_1) - \frac{e_{c_1} (\theta_2 - \theta_1)^2}{2} \qquad (2\text{-}9)$$

observing that $e_{s_2} = \sin \theta^*_2 - \sin \theta_2$

and $e_{c_2} = \cos \theta^*_2 - \cos \theta_2$

Equations (2-8) and (2-9) can be rewritten as follows:

$$e_{s_2} - e_{s_1} = e_{c_1} (\theta_2 - \theta_1) - \frac{e_{s_1} (\theta_2 - \theta_1)^2}{2} \qquad (2\text{-}10)$$

$$e_{c_2} - e_{c_1} = -e_{s_1} (\theta_2 - \theta_1) - \frac{e_{c_1} (\theta_2 - \theta_1)^2}{2} \; . \qquad (2\text{-}11)$$

Equations (2-10) and (2-11) can be approximated by the differential equations

$$de_s = e_c d\theta - \frac{e_{s_1} d\theta^2}{2}$$

$$de_c = -e_s d\theta - \frac{e_{c_1} d\theta^2}{2} \; .$$

The $d\theta^2$ terms are negligible, as compared to the others, thus

$$de_s = e_c d\theta \tag{2-12}$$

$$de_c = -e_s d\theta \quad . \tag{2-13}$$

where the equations of a sine-cosine generator are recognized. The solutions are

$$e_s = C \sin(\theta + \phi) \tag{2-14}$$

$$e_c = C \cos(\theta + \phi) \quad . \tag{2-15}$$

2.3.2. Error Prediction For The Free-Running Sine-Cosine Generator. Equations (2-14) and (2-15) show that, for a given initial error, e_s, and corresponding e_c in quadrature, the errors will propagate as two sinewaves in quadrature for e_s and e_c, without change of the peak amplitude, C. However, the problem conditions are different since initial errors are not necessarily in quadrature and source errors introduce constantly new initial errors in a more or less random manner.

It means that e_s and e_c should consist of two nearly sinewaves in quadrature, where e_s leads e_c, the basic frequency being that of the sine-cosine generator. However, there will be phase and amplitude variations corresponding to the irregularity of the truncation error. Over a sufficient number of revolutions, the error curve (near sine wave) amplitude will fluctuate, with an expected value corresponding to the average error source amplitude computed in Section 2.2.5 to be 38 pulses after 20 revolutions. The peak amplitude fluctuation limits are zero and worst-case error. The worst-case error in this example is

$$e_T \text{ (smoothing)} = \xi_{max} + \epsilon + e_{SR\ max} = \frac{(0.9)\,(17)\,(80) + 6 + 40}{4} = 320 \text{ pulses}.$$

And for 2-bit smoothing; total source error (smoothing)

$$\bar{e}_T(\text{smoothing}) < \frac{\xi + \epsilon + e_{SR}}{4} \qquad (2\text{-}19)$$

or

$$\bar{e}_T(\text{smoothing}) < 29 \text{ pulses} \quad .$$

Worst-case error peak amplitude is computed by linear addition of truncation and sign-reversal errors and the integrand quantization error remains unchanged.

$$\bar{e}_T \text{ (smoothing)} < \frac{(0.7)\ (17)\ (80) + 3 + 40}{4} = 250 \text{ pulses} \quad . \qquad (2\text{-}20)$$

2.4 Direction Cosines – Error Prediction In the General Case of 3-Angle Variations

2.4.1 Error Sources. The error sources are identical to those found in the sine-cosine generator. The following program will be analyzed. The driving angles are:

$\theta_1 = 1/2\ \theta_2$

θ_2 = Reference

$\theta_3 = 1/4\ \theta_2$

θ_2 is driven from 0 to 130 radians. An inspection of the solution shows that a_1, a_2, a_3 are periodic functions of equal frequency, similar to sine waves and with a 120° phase shift from one to another. (Program Run 3.108)

a_1 varies from -0.65 to + 0.9 of full scale; $\bar{a}_1 \simeq +0.1$

a_2 varies from -0.2 to +0.63 of full scale; $\bar{a}_2 \simeq +0.2$

a_3 varies from -0.8 to + 0.9 of full scale; $\bar{a}_3 \simeq +0.05$.

There are 23 revolutions of the direction cosines, where θ_2 is driven from 0 to 125 radians. The error sources will be

(a) Truncation Errors

$$\xi_{a_1} = \sqrt{(17)^2\ [\,(0.65)^2 + (0.9)^2]\ 46} = 130 \text{ pulses} \qquad (2\text{-}21)$$

$$\xi_{a_2} = \sqrt{(17)^2\ [\,(0.2)^2 + (0.63)^2]\ 46} = 77 \text{ pulses}$$

$$\xi_{a_3} = \sqrt{(17)^2\ [\,(0.8) + (0.9)^2]\ 46} = 142 \text{ pulses}$$

(b) Integrand random quantization errors

$$\epsilon_{a_1, \theta_2} = \frac{1}{\sqrt{12}} \quad \frac{125}{\sqrt{(48)(1.55)}} = 4.2 \text{ pulses} \qquad (2\text{-}22)$$

$$\epsilon_{a_1, \theta_3} = \frac{1}{\sqrt{12}} \quad \frac{31}{\sqrt{(48)(1.55)}} = 1 \text{ pulse}$$

$$\epsilon_{a_2, \theta_1} = \frac{1}{\sqrt{12}} \quad \frac{62}{\sqrt{(48)(0.83)}} = 3 \text{ pulses}$$

$$\epsilon_{a_2, \theta_3} = \frac{1}{\sqrt{12}} \quad \frac{31}{\sqrt{(48)(0.83)}} = 1.5 \text{ pulses}$$

$$\epsilon_{a_3, \theta_1} = \frac{1}{\sqrt{12}} \quad \frac{62}{\sqrt{(48)(1.7)}} = 2 \text{ pulses}$$

$$\epsilon_{a_3, \theta_2} = \frac{1}{\sqrt{12}} \quad \frac{125}{\sqrt{(48)(1.7)}} = 4 \text{ pulses} \quad .$$

(c) Integrand sign-reversal errors

$e_{SR} = \sqrt{46} \simeq 7$ pulses for each multiplier.

(d) Total source error

$$\bar{e}_T (a_1 \text{ input}) = 1.5 + 4 + 14 + 77 + 142 = 238.5 \text{ pulses} \qquad (2\text{-}23)$$

$$\bar{e}_T (a_2 \text{ input}) = 2 + 1 + 14 + 130 + 142 = 289 \text{ pulses}$$

$$\bar{e}_T (a_3 \text{ input}) = 4.2 + 3 + 14 + 130 + 77 = 228.2 \text{ pulses}$$

and assuming a 2-bit smoothing

$$\bar{e}_{T \text{ smoothing}} (a_1) = 60 \text{ pulses} \qquad (2\text{-}24)$$

$$\bar{e}_{T \text{ smoothing}} (a_2) = 72 \text{ pulses}$$

$$\bar{e}_{T \text{ smoothing}} (a_3) = 57 \text{ pulses} \quad .$$

2.4.2 Error Propagation. An inspection of the DO direction-cosine program makes an evaluation possible of the error-propagation equations similar to those for the sine-cosine generator described in Section 2.3.1. Let a_{11}, a_{21}, a_{31} be the values of direction cosines at the beginning of a computation interval, and a_{12}, a_{22}, a_{32} the values of the same direction cosines at the end of the computation interval. The asterisk (a_{12}^*) indicates the DO value and no asterisk indicates the desired value. $e_{a_{11}}$, $e_{a_{12}}$ ... etc., designate the errors on the value of a_1 at the beginning and end of the computation interval. Only the error propagation is evaluated; error sources will not be introduced into the equations.

$$a^*_{12} = a_{11} + e_{a_{11}} + \int_{\theta_{31}}^{\theta_{32}} [\, a_{21} + e_{a_{21}} + \int_{\theta_{11}}^{\theta_1} (a_3 + e_{a_{31}})\, d\theta_1 - \int_{\theta_{31}}^{\theta_3} (a_1 + e_{a_{11}})\, d\theta_3]\, d\theta_3$$

$$- \int_{\theta_{21}}^{\theta_{22}} [\, a_{31} + e_{a_{31}} + \int_{\theta_{21}}^{\theta_2} (a_1 + e_{a_{11}})\, d\theta_2 - \int_{\theta_{11}}^{\theta_1} (a_2 + e_{a_{21}})\, d\theta_1]\, d\theta_2 \quad .$$

After reduction and elimination of the second order terms,

$$a^*_{12} - a_{12} = e_{a_{11}} + e_{a_{21}} (\theta_{32} - \theta_{31}) - e_{a_{31}} (\theta_{22} - \theta_{21})$$

similarly for the other direction cosines

$$a^*_{12} - a_{22} = e_{a_{21}} + e_{a_{31}} (\theta_{12} - \theta_{11}) - e_{a_{11}} (\theta_{32} - \theta_{31})$$

$$a^*_{32} - a_{32} = e_{a_{31}} + e_{a_{11}} (\theta_{22} - \theta_{21}) - e_{a_{21}} (\theta_{12} - \theta_{11}) \quad .$$

Using the approximation of differential equations, it results in

$$d(e_{a_1}) = e_{a_2}\, d\theta_3 - e_{a_3}\, d\theta_2 \qquad (2\text{-}25)$$

$$d(e_{a_2}) = e_{a_3}\, d\theta_1 - e_{a_1}\, d\theta_3$$

$$d(e_{a_3}) = e_{a_1}\, d\theta_2 - e_{a_2}\, d\theta_1$$

where the equations of the direction cosines are recognized. The interpretation will be similar to that developed for the sine-cosine generator. For a given

set of initial errors, e_{a_1}, e_{a_2}, e_{a_3}, the errors will propagate like the three direction cosines without change in amplitude. In the problem conditions, where irregular truncation errors are generated throughout the computation, the basic frequency of the three error signals will vary, although their relative phase shift will remain close to 120°. The peak amplitudes grow irregularly with θ; the expected peak amplitudes will be, after a variation of θ_2, from 0 to 125 radians.

$$\bar{e}_T(a_1) = 60 \text{ pulses}$$

$$\bar{e}_T(a_2) = 72 \text{ pulses} \qquad (2\text{-}26)$$

$$\bar{e}_T(a_3) = 57 \text{ pulses} \quad .$$

Similarly to the direction cosines themselves, these errors will be biased in one direction. However, the statistical approach to the error prediction does not permit a prediction of the sign and amplitude of this bias. An upper limit on this bias error is known, since Eq. (2-26) includes this bias value. The worst-case error peak amplitude will be

$$e_T(\text{smoothing, } a_3 \text{ input}) = \frac{4.2 + 3 + 2\ (46) + \sqrt{46}\,(130 + 77)}{4} = 375 \text{ pulses}. \qquad (2\text{-}27)$$

2.4.3. Error Prediction for Limit Cycles of the Direction-Cosine Generator. The discussion of Section 2.3.3, in the case of the sine-cosine generator is also applicable here. A reduction of the error may occur due to partial reversibility of the DO multiplier circuit. Magnitude of the error reduction is not possible at this time on an analytical basis; however, a limit of the average error magnitude can be predicted.

Let's choose the example of the direction-cosine generator where θ_1, θ_2 θ_3 are driven at the same rate. The source errors after 20 cycles of ± 1 radian amplitude will be estimated as follows

$$\xi_{a_1} = \sqrt{(0.5)^2 (4)(20)(17)^2} = 76 \text{ pulses} \tag{2-28}$$

$$\epsilon_{a_i \theta_j} = \frac{1}{\sqrt{12}} \frac{80}{\sqrt{(2)(20)}} = 3.7 \text{ pulses} \tag{2-29}$$

$$e_{SR}(a_i) = \sqrt{(2)(20)} = 6.3 \text{ pulses} \quad . \tag{2-30}$$

The total error source will be

$$\bar{e}_T(a_i) = 2(76 + 3.7 + 6.3) = 172 \text{ pulses} \tag{2-31}$$

and if a two-bit smoothing is used

$$\bar{e}_T(a_i \text{ smoothing}) = \frac{172}{4} = 43 \text{ pulses} \quad . \tag{2-32}$$

Therefore, if some reduction of the error is introduced through partial reversibility, Eq. (2-32) becomes

$$\bar{e}_T(\text{smoothing}, a_i) < 43 \text{ pulses} \quad . \tag{2-33}$$

The error curve equations, being similar to the direction-cosine equations, will consist of segments of three biased, near sine waves with a 120° phase shift from one to another.

The worst-case error peak amplitude will be:

$$\bar{e}_T(\text{smoothing}, a_3) = \frac{2(76\sqrt{80} + 3.7 + 40)}{4} = 362 \text{ pulses} \quad . \tag{2-34}$$

2.5. Comparison of Error Prediction and Experimental Data

This comparison will be best conducted by setting up the following table.

$$\xi_{a_1} = \sqrt{(0.5)^2 (4)(20)(17)^2} = 76 \text{ pulses} \tag{2-28}$$

$$\epsilon_{a_i \theta_j} = \frac{1}{\sqrt{12}} \frac{80}{\sqrt{(2)(20)}} = 3.7 \text{ pulses} \tag{2-29}$$

$$e_{SR}(a_i) = \sqrt{(2)(20)} = 6.3 \text{ pulses} \quad . \tag{2-30}$$

The total error source will be

$$\bar{e}_T(a_i) = 2(76 + 3.7 + 6.3) = 172 \text{ pulses} \tag{2-31}$$

and if a two-bit smoothing is used

$$\bar{e}_T(a_i \text{ smoothing}) = \frac{172}{4} = 43 \text{ pulses} \quad . \tag{2-32}$$

Therefore, if some reduction of the error is introduced through partial reversibility, Eq. (2-32) becomes

$$\bar{e}_T(\text{smoothing}, a_i) < 43 \text{ pulses} \quad . \tag{2-33}$$

The error curve equations, being similar to the direction-cosine equations, will consist of segments of three biased, near sine waves with a 120° phase shift from one to another.

The worst-case error peak amplitude will be:

$$\bar{e}_T(\text{smoothing}, a_3) = \frac{2(76\sqrt{80} + 3.7 + 40)}{4} = 362 \text{ pulses} \quad . \tag{2-34}$$

2.5. Comparison of Error Prediction and Experimental Data

This comparison will be best conducted by setting up the following table.

Table 2-1. Comparison of Error Prediction and Experimental Data

PROBLEM RUN	PREDICTED ERROR	EXPERIMENTAL ERROR
		Program Run 1.101
Sine-Cosine Generator	(a) 2 Pseudo-sinewaves in quadrature of increasing amplitude at a frequency $f \simeq f_o$.	(a) 2 Pseudo-sinewaves in quadrature of increasing amplitude at a frequency $f \simeq f_o$.
Free-running at frequency f_o for 20 revolutions	(b) Small variations of f.	(b) Small variations of f.
2-bit smoothing	(c) Irregular peak amplitude growth.	(c) Irregular peak amplitude growth.
	(d) Expected peak amplitude after 20 revolutions:	(d) Peak amplitude after 20 revolutions:
	$e_p \sin\theta \simeq e_p \cos\theta = 38$ pulses	$e_p \sin\theta \simeq e_p \cos\theta = 47$ pulses
	(e) Maximum peak amplitude:	
	e_p max = 320 pulses	
		Program Run 1.108
Sine-Cosine Generator	(a) 2 segments of pseudo-sine waves in quadrature and of increasing amplitude.	(a) 2 segments of pseudo-sinewaves in quadrature and of increasing amplitude.
Limit Cycles ± 1 Radian, 20 cycles basic frequency f_o.	(b) Small phase variations.	(b) Error curves are biased. and quite irregular; peak amplitude of the error at 10 cycles = 1.5 times peak error amplitude at 20 cycles.
2-bit smoothing	(c) Peak amplitude variations.	(c) Peak amplitude after 20 cycles
	(d) Expected peak amplitude after 20 cycles.	$e_p \theta \simeq e_p \cos\theta = 8$ pulses.

Table 2-1. Comparison of Error Prediction and Experimental Data (Cont)

PROBLEM RUN	PREDICTED ERROR	EXPERIMENTAL ERROR
	$e_p \sin\theta \simeq e_p \cos\theta <$ 29 pulses (e) Maximum peak amplitude $e_{p\,max} < 250$ pulses.	
		Program Run 3.108
Direction-Cosine Generator	(a) Three 120° phase-shifted biased pseudo-sinewaves of increasing amplitude, at a frequency $f \simeq f_o$.	Three 120° phase-shifted biased pseudo-sinewaves of increasing amplitude, at a frequency $f \simeq f_o$.
Free running at frequencies: f_0 for θ_2 $f_0/2$ for θ_1 $f_0/4$ for θ_3	(b) Small variations of f.	(b) Small variations of f.
a_1, a_2, a_3 are 120° phase-shifted, positive-biased sinewaves of frequency f_o (23 cycles for a variation of θ_2 of 125 radians).	(c) Irregular peak amplitude growth.	(c) Irregular peak amplitude growth.
2-bit smoothing.	(d) Biased errors. (e) Expected peak amplitude after $\theta_2 = 125$ radians. $\bar{e}_p(a_1) = 61$ pulses $\bar{e}_p(a_2) = 74$ pulses $\bar{e}_p(a_3) = 58$ pulses	(d) Errors negatively biased. (e) Peak amplitude after $\theta_2 = 125$ radians. $e_p(a_1) = 61$ pulses $e_p(a_2) = 54$ pulses $e_p(a_3) = 61$ pulses

Table 2-1. Comparison of Error Prediction and Experimental Data (Cont)

PROBLEM RUN	PREDICTED ERROR	EXPERIMENTAL ERROR
	(f) Maximum peak amplitude $e_p(a_3) = 375$ pulses	(f) Maximum peak amplitude $e_p(a_3) = 215$ pulses (Data Run - 3-208)
		Program Run 3.103
Direction-Cosine Generator	(a) Segments of three 120° phase-shifted biased pseudo-sine waves of increasing amplitude at a frequency $f \simeq f_o$.	(a) Very irregular error curves somewhat 120° phase shifted.
Limit cycles $\theta_2 \pm 1$ radian, 20 cycles	(b) Small phase variations.	(b) Large negative bias increasing with peak amplitude and with θ_2.
a_1, a_2, a_3 are segments of the free-running direction-cosine curves.	(c) Irregular peak amplitude growth.	(c) Peak amplitude after 20 cycles: $e_p(a_1) = 23$ pulses $e_p(a_2) = 27$ pulses $e_p(a_3) = 17$ pulses
2-bit smoothing	(d) Expected peak amplitude after 20 cycles $\bar{e}_p(a_1) = \bar{e}_p(a_2) = \bar{e}_p(a_3)$ < 43 pulses	
	(e) Maximum peak amplitude $e_p(a_1) < 362$ pulses	

2.6. Reversible Computation For Small Limit Cycles

The semi-statistical approach to the error analysis does not permit reversibility prediction of the computation, however, as discussed in 2.3.3 at least partial reversibility can be expected. Complete reversibility means that, starting from an initial condition for the three direction-cosines a_{10}, a_{20}, a_{30}, and the three angles, θ_{10}, θ_{20}, θ_{30}, the direction-cosine counters will return to the same values, a_{10}, a_{20}, a_{30}, after any variations of the angles θ_1, θ_2, θ_3, when these angles come back simultaneously to the initial value θ_{10}, θ_{20}, θ_{30}. Simple experiments proved that for large angle variations, there is no complete reversibility for the DO implementation of the direction-cosine problem, however, partial and perfect reversibility have been observed for small angle variations in the following conditions:

Data Run 3.105 (direction-cosine): The three angles, θ_1, θ_2, θ_3, are driven simultaneously on constant amplitude cycles of ± 1/64 radian amplitude. The values of the direction-cosines, a_1, a_2, a_3, repeat exactly at each new cycle (Table 2.2).

Data Run 1.110 (sine-cosine): The angle θ is driven randomly positive and negative with a limit maximum amplitude, θ_m; there are approximately 30 sign reversals of θ per machine run. θ is then driven to the initial value, θ_o, and the final values of sin θ and cos θ are compared to the initial value. The values of sin θ and cos θ repeat exactly in the following conditions:

θ_m	θ_o
1/64 radian	0 radian
1/32 radian	3.5 radian
1/32 radian	5.25 radian
1/32 radian	5.75 radian

For other initial conditions, there is no exact repeatability for angles larger than 1/128 radian; however, the errors remain small. Repeatability has not been checked for angles less than 1/128 radians.

Table 2.2

Program Run 3.105

Limit cycle $\Delta\theta_3 = \Delta\theta_2 = \theta_1 = 1/64$ radian

Cycle	$\Delta\theta$	a_1			ϵ	a_2			ϵ	a_3			ϵ
1	0	+01111	01101	01101	0	-00100	10010	01011	0	+10111	11100	10001	0
1				01101	0			01010	-1			10001	0
2				01101	0			01011	0			10001	0
2				01101	0			01010	-1			10001	0
2				01101	0			01011	0			10001	0
10				01101	0			01011	0			10001	0
10				01101	0			01010	-1			10001	0
10				01101	0			01011	0			10001	0
20				01101	0			01011	0			10001	0
20				01101	0			01010	-1			10001	0
20				01101	0			01011	0			10001	0
1	+1/64	+01110	11111	00100	0	-00100	01101	11100	+1	+11000	00110	01011	-1
2				00100	0			11100	+1			01011	-1
10				00100	0			11100	+1			01011	-1
20				00100	0			11100	+1			01011	-1
1	-1/64	+01111	11011	10010	0	-00100	10110	01100	0	+10111	10010	01100	0
2				10010	0			01100	0			01100	0
10				10010	0			01100	0			01100	0
20				10010	0			01100	0			01100	0

Results from these experiments demonstrate that the computation is either reversible or near reversible for small angle variations. More data would be desirable to refine the study of reversibility in the case of small random variations of the angles as it is an important feature for attitude computation and similar applications.

3.0 MEDOC SIMULATION, TASK II

MEDOC, a Modular-Expandable, Digital-Operational Computer breadboard, is illustrated in the photograph, Figure 3.1. The machine consists of a computer rack and control rack. The computer section contains four drawers that can accommodate five, 20-bit counters per drawer. Space is also available to plug-in other operational blocks such as pulse multipliers, pulse adders, delay, and decision units. The control section contains an input keyboard for purposes of resetting and initializing the computer, a clock-slot generator and control circuitry for printout, and program control. In addition, a D/A converter, X-Y plotter is available for connection to the computer for graphical outputs. The computer program is patched by connecting cables between operational blocks at terminals on the right-front side of the computer rack. Scaling is accomplished by plug-in connectors, which are located on the right side, and clock and data inputs which are connected to the left-front side of the computer rack. The circuitry is designed for 5-Mc operation.

3.1 Statement of the Problem

The attitude computation in a typical strap-down system is illustrated in Figure 3.2. It is assumed that the gyro resolution is between 2^{-11} and 2^{-20} radians per pulse and a maximum sensing rate of 60 degrees per second. The integrating accelerometers were assumed to have a 0.1 foot per second per pulse resolution.

The incremental changes of the direction cosines are related by the following equations:

$$\begin{aligned} \Delta a_1 &= a_2 \Delta\theta_3 - a_3 \Delta\theta_2 \\ \Delta a_2 &= a_3 \Delta\theta_1 - a_1 \Delta\theta_3 \\ \Delta a_3 &= a_1 \Delta\theta_2 - a_2 \Delta\theta_1 \end{aligned} \qquad (3\text{-}1)$$

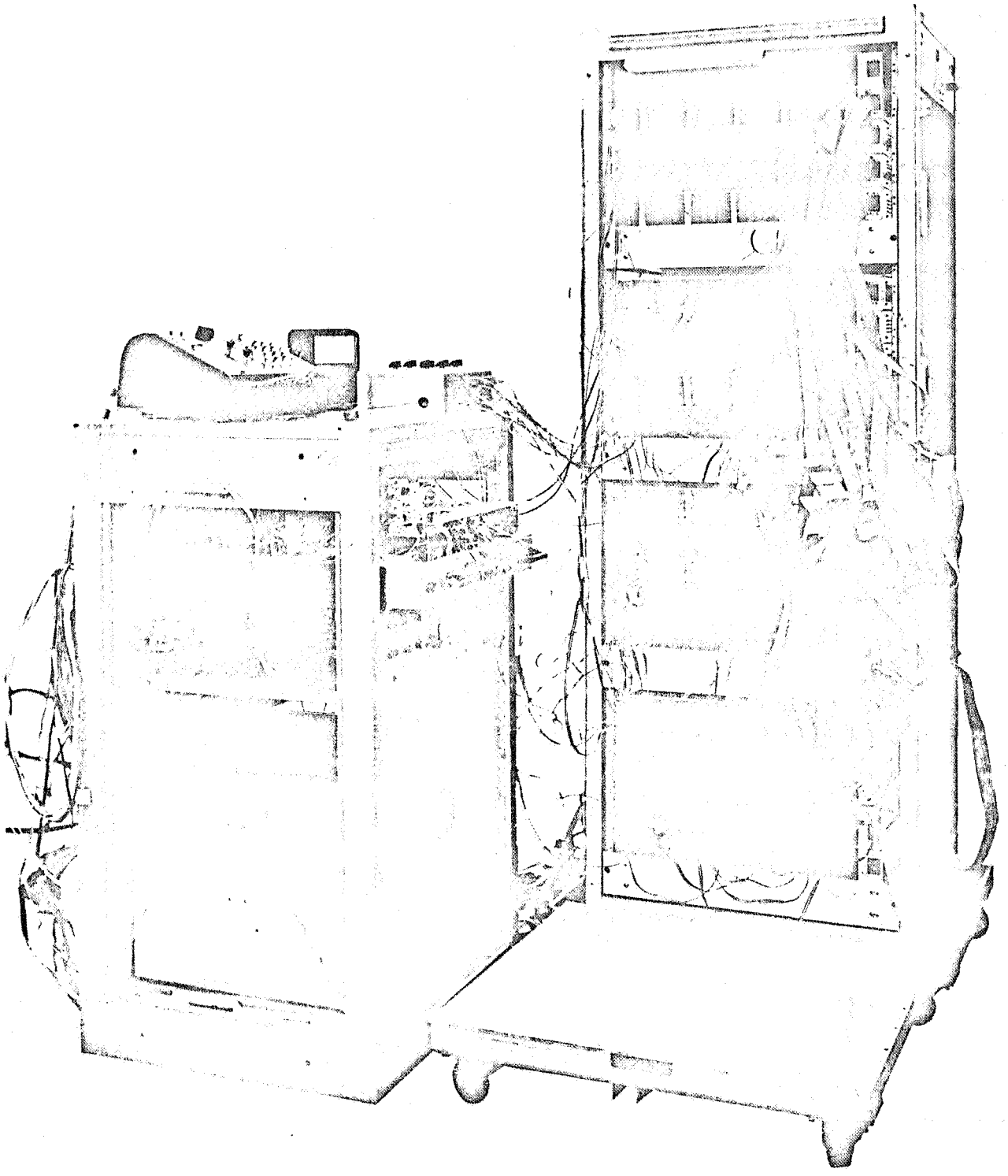


Figure 3.1 MEDOC (Modular-Expandable, Digital-Operational Computer) Breadboard

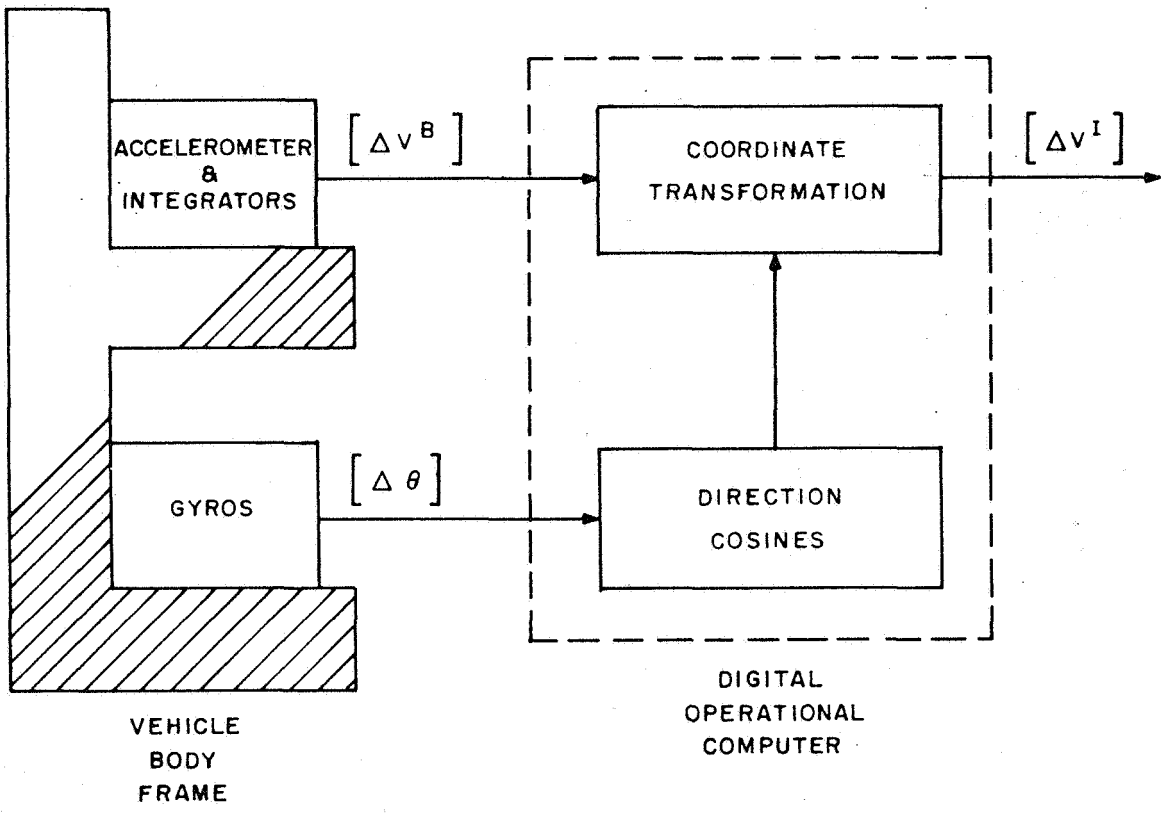


Figure 3.2 Strap-Down System Attitude Computation

$$\Delta b_1 = b_2\Delta\theta_3 - b_3\Delta\theta_2$$

$$\Delta b_2 = b_3\Delta\theta_1 - b_1\Delta\theta_3 \qquad (3\text{-}2)$$

$$\Delta b_3 = b_1\Delta\theta_2 - b_2\Delta\theta_1$$

$$\Delta c_1 = c_2\Delta\theta_3 - c_3\Delta\theta_2$$

$$\Delta c_2 = c_3\Delta\theta_1 - c_1\Delta\theta_3 \qquad (3\text{-}3)$$

$$\Delta c_3 = c_1\Delta\theta_2 - c_2\Delta\theta_1$$

The transformation from body frame to inertial reference frame, of the velocity vector ΔV^B, is given by:

$$\begin{bmatrix} V_x^I \\ V_y^I \\ V_z^I \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} V_x^B \\ V_y^B \\ V_z^B \end{bmatrix}$$

or

$$\begin{aligned} \Delta V_x^I &= a_1\Delta V_x^B + a_2\Delta V_y^B + a_3\Delta V_z^B \\ \Delta V_y^I &= b_1\Delta V_x^B + b_2\Delta V_y^B + b_3\Delta V_z^B \\ \Delta V_z^I &= c_1\Delta V_x^B + c_2\Delta V_y^B + c_3\Delta V_z^B \end{aligned} \qquad (3\text{-}4)$$

.

3.2 MEDOC Program

As far as the MEDOC program runs are concerned, it was only necessary to program the set of equations in (3-1) and the first equations of set (3-4). This is true, since they are independent of the other equations. Conceptually, three sequential machine runs with the identical input variables and new initial conditions for the direction-cosines and velocity equations will simulate the total set of equations. Figure 3.3 illustrates the MEDOC program. Note that only three counters are needed for the three direction-cosine generators. The three $\Delta\theta$ counters (independent variable) are shared with the direction-cosine counters. Thus, for nine direction-cosines, only 12 counters are needed as opposed to the DDA requirement of 18 registers.

Program sets numbered 1.000 through 4.000 were generated to investigate one angle, two angle, three angle, and velocity inputs. Tables B.1 through B.8 in Appendix B summarize the results of the program runs. Appendix B also contains the pertinent error curves as discussed in this report. The range and resolution of variables are as follows:

$$0 \leq a_i \leq 1.0$$

$$0 \leq \theta_i \leq 126 \text{ radians}$$

$$0 \leq V_x^I \leq 3276.8 \text{ ft./sec.}$$

$$\Delta\theta_i = 2^{-15} \text{ radians}$$

$$\Delta V_i^B = 0.1 \text{ ft. sec.}$$

The MEDOC machine units (mu) are:

$[\theta]$ mu = 1 radian

$[V]$ mu = 3276.8 ft/sec.

n = basic scale = 15 bits

Pulse Weight $\begin{cases} = 2^{-15} \text{ radians} \\ = 0.1 \text{ ft./sec.} \end{cases}$

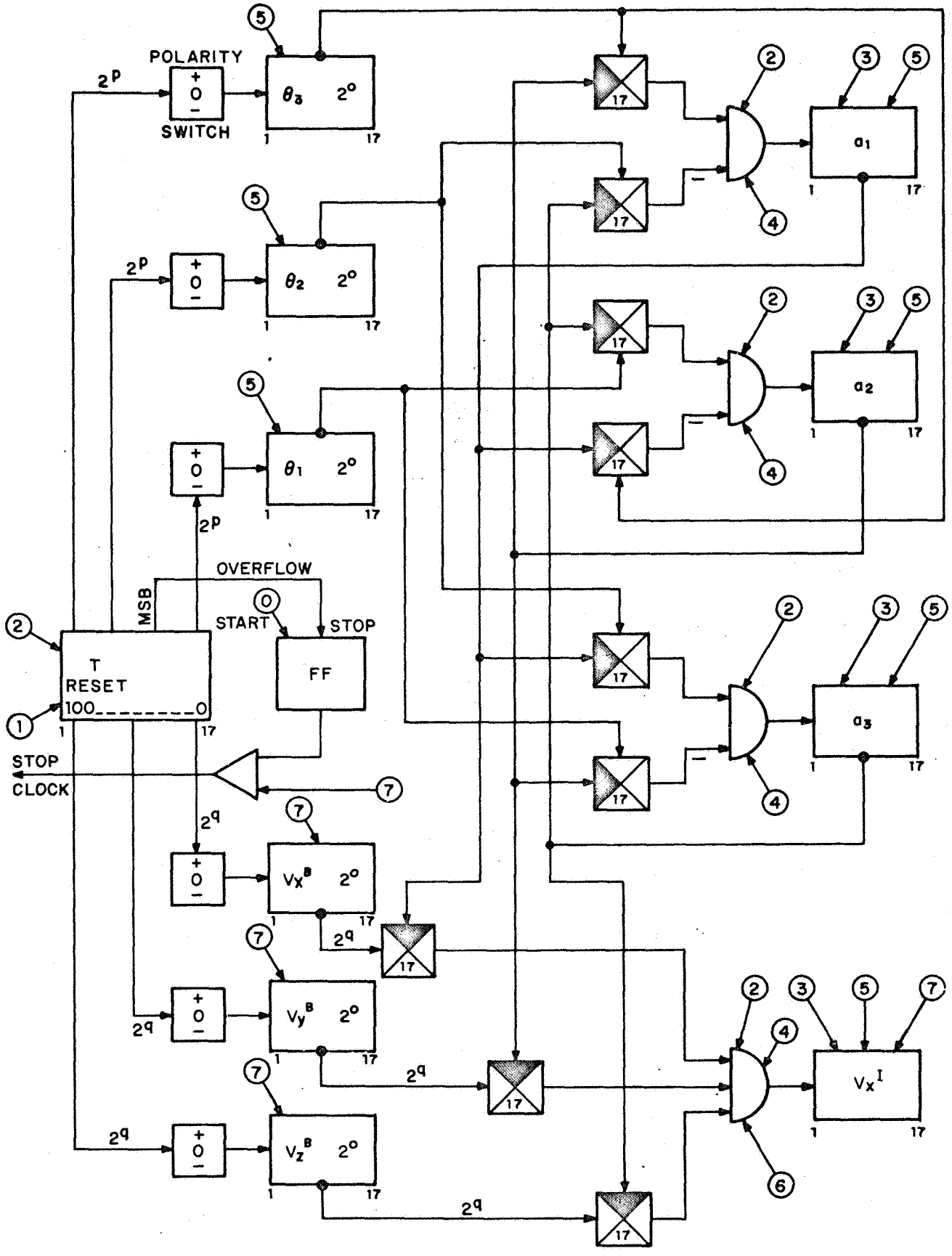


Figure 3.3 MEDOC Program

It should be obvious to the reader that, due to time considerations, an exhaustive number of program runs to investigate the effects of changes in initital conditions and relative data input rates is an impossibility. Thus, it became necessary to select a sufficient number of typical programs based upon past experience. For example, it is known that the DO technique is more "error sensitive" to initial conditions around mid-scale of a counter; hence, initial-condition considerations were concerned with this fact. It is also known that the relative rate between the code and variable input of the pulse multiplier is a factor which affects the error curve, and this too was a parameter to be programmed. Basically, two types of programs were run, insofar as the direction-cosine computer was concerned. First, rotations both clockwise (+) and counterclockwise (-), up to 126 radians were run. Secondly, limit cycles with amplitude variations from 2^{-15} to 2 radians up to 20 cycles were also run. Insofar as the velocity runs were concerned, attention was directed to programming velocities in terms of ascent missions. Since only a simple multiplication and addition is required, Eq. (3-4), not many programs of this type were run. Program set #1 contained 11; program set #2 contained 9; program set #3 contained 20, and program set #4 contained 6 program runs for a total of 46. Appendix B summarizes and tabulates the results of all the program runs. Error curves pertinent to this report are also included. Copies of those error curves which are not included herein can be obtained by writing to the Program Manager. Their usefulness is evident when comparing the DO technique with results of other special-purpose computer techniques.

3.3 Scaling and Smoothing

The important element in a digital-operational system is the pulse multiplier. This operational unit is designed to scale down an input pulse train that is equal to the product of the number of input pulses (multiplicand) and a numeric code (multiplier). If a pulse train of frequency f is applied to the first binary scaler so that x pulses occur in a given time interval, the scaler outputs of each binary during the interval in which x input pulses arrive are x/2, x/4, x/8.....x/2n pulses. The sum can be expressed as:

$$X \sum_{1}^{n} \frac{1}{2n} = 1 - \frac{1}{2n} \tag{3-5}$$

$$\sum_{1}^{\infty} \frac{1}{2^n} = 1 \quad . \tag{3-6}$$

Therefore, a scalar output must occur for each input pulse. These scaler outputs are applied to gates which are opened or closed according to the multiplier y, where

$$Y = \sum_{1}^{n} \frac{a_n}{2^n} \quad , \quad n \text{ is an integer} \quad . \tag{3-7}$$

The outputs of the gates are combined to yield a sum of pulses

$$a_1 \frac{x}{2} + a_2 \frac{x}{4} + \ldots\ldots a_n \frac{x}{2^n} \tag{3-8}$$

which is actually the product of xy

$$Z = XY = \sum_{1}^{n} \frac{X\, a_n}{2^n} = X \sum_{1}^{n} \frac{a_n}{2^n} \quad . \tag{3-9}$$

It should be recognized that y is always less than unity.

The relationship, Equation (3-9), is exact as indicated when x is a multiple of 2^n and when y is a constant over that interval. If y changes slowly, compared to the pulse rate of x, the multiplication is approximately correct. With reference to the timing diagram, Figure 3.4, it can be shown that the output pulses are asynchronous, and instantaneous errors exist during the binary timing interval. For example, if we consider a 4-stage multiplier and y = 10, then an output of 10 pulses results for every 16 pulses on the input.

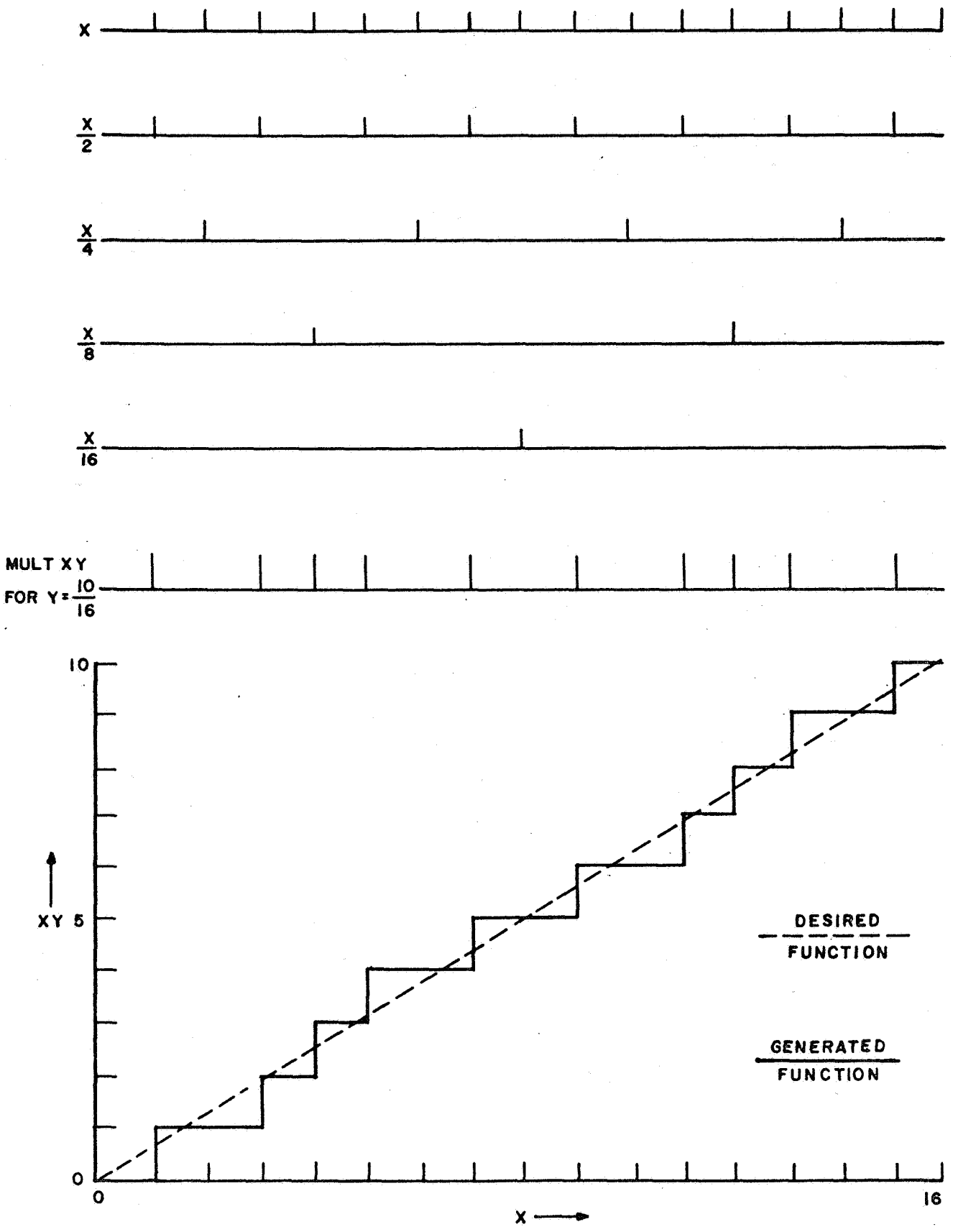


Figure 3.4 Pulse Multiplier Timing Diagram

To improve the asynchronous performance of the multiplier, the input frequency may be increased by some binary multiple and then the output divided by this same multiple, Figure 3.5.

The improvement in accuracy in the order of three due to smoothing is demonstrated in Figure 3.6. This figure illustrates the error curve of the sine-cosine generator during a limit cycle program (Program Run 1.108) of ± 1 radian amplitude over a period of 20 cycles. Smoothing was accomplished by upscaling the input angular pulses by a factor of four, and then dividing by four when reading the direction-cosine counters. The factor of four was selected as an optimum choice based upon past experience. A factor of eight would not increase the accuracy sufficiently, compared to a factor of four, to warrant the additional bit in each counter. Thus, it was decided that smoothing was warranted and the additional bits were added to the basic counter scale. It was also decided to scale each of the direction-cosine and initial-velocity counters to 0.9 to prevent overflow.

3.4 Initial Conditions

The computation for initial conditions was based upon the well-known relationship between the direction-cosines and the Euler angle transformations:

$$
\begin{aligned}
a_1 &= C\,\emptyset\,C\gamma \\
a_2 &= C\,\emptyset\,S\gamma\,S\psi - S\,\emptyset\,C\psi \\
a_3 &= C\,\emptyset\,S\gamma\,C\psi + S\,\emptyset\,S\,\psi
\end{aligned}
\qquad (3\text{-}10)
$$

where $\emptyset$ (Yaw), γ (pitch), and ψ (roll) represent the Eulerian angles and a_1, a_2, and a_3 are the direction cosines for X^B, one of the body axis coordinates. Thus, Eulerian angle initial conditions were assumed, the direction cosines computed, and the MEDOC initialized to these values.

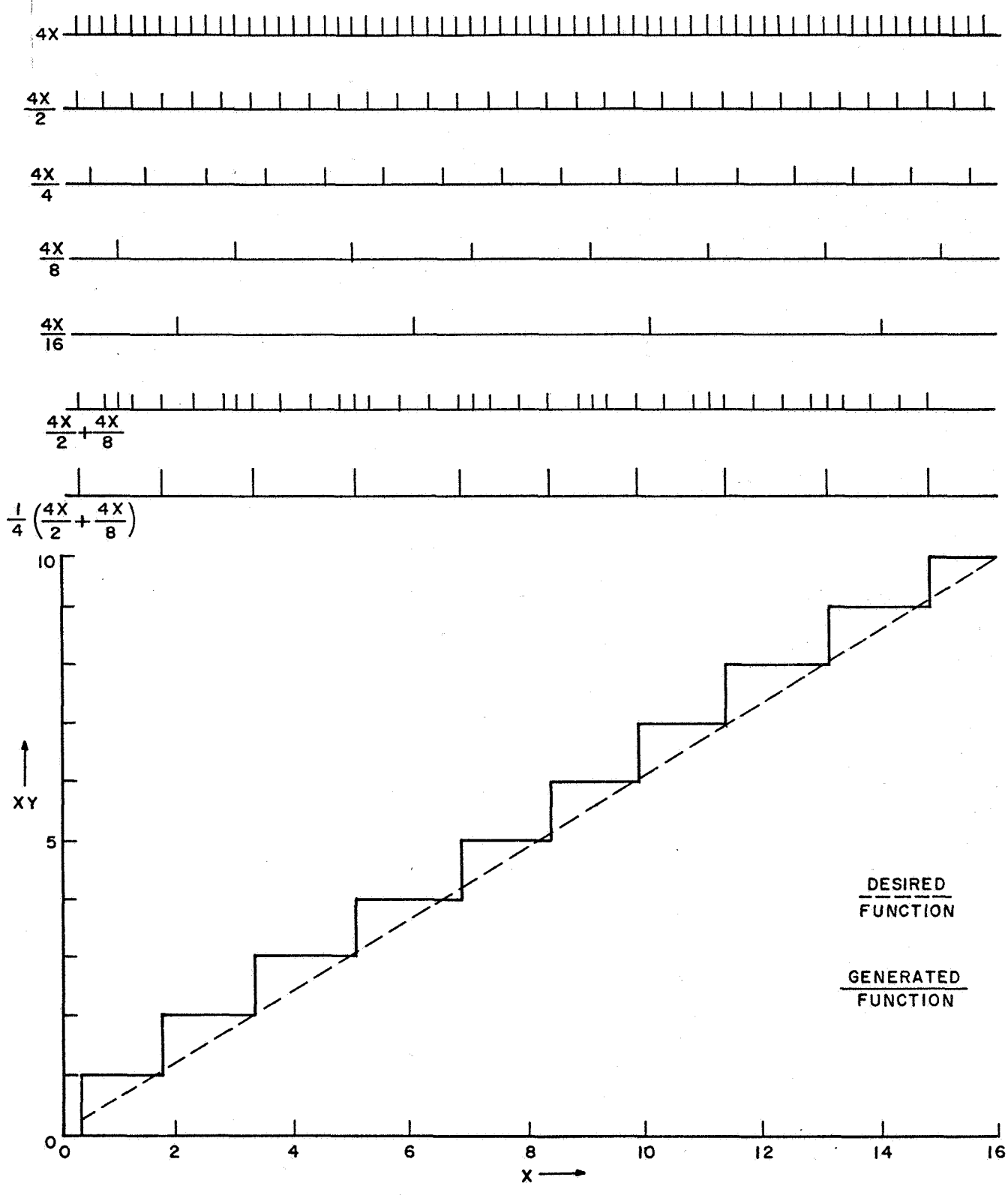


Figure 3.5 Pulse Multiplier Output with Digital Smoothing

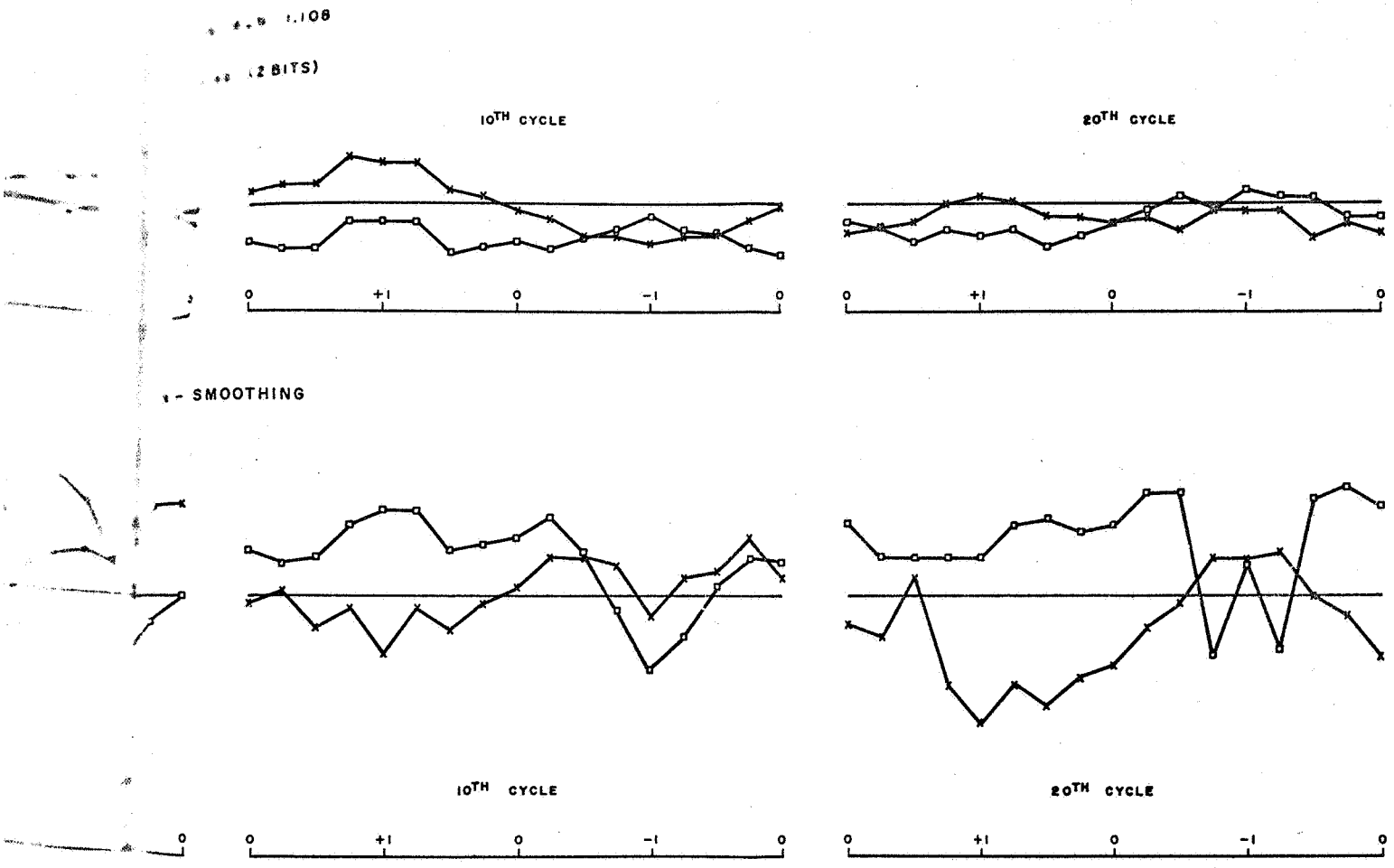


Figure 3.6 Sine-Cosine Error Curve – Smoothing vs. Non-Smoothing

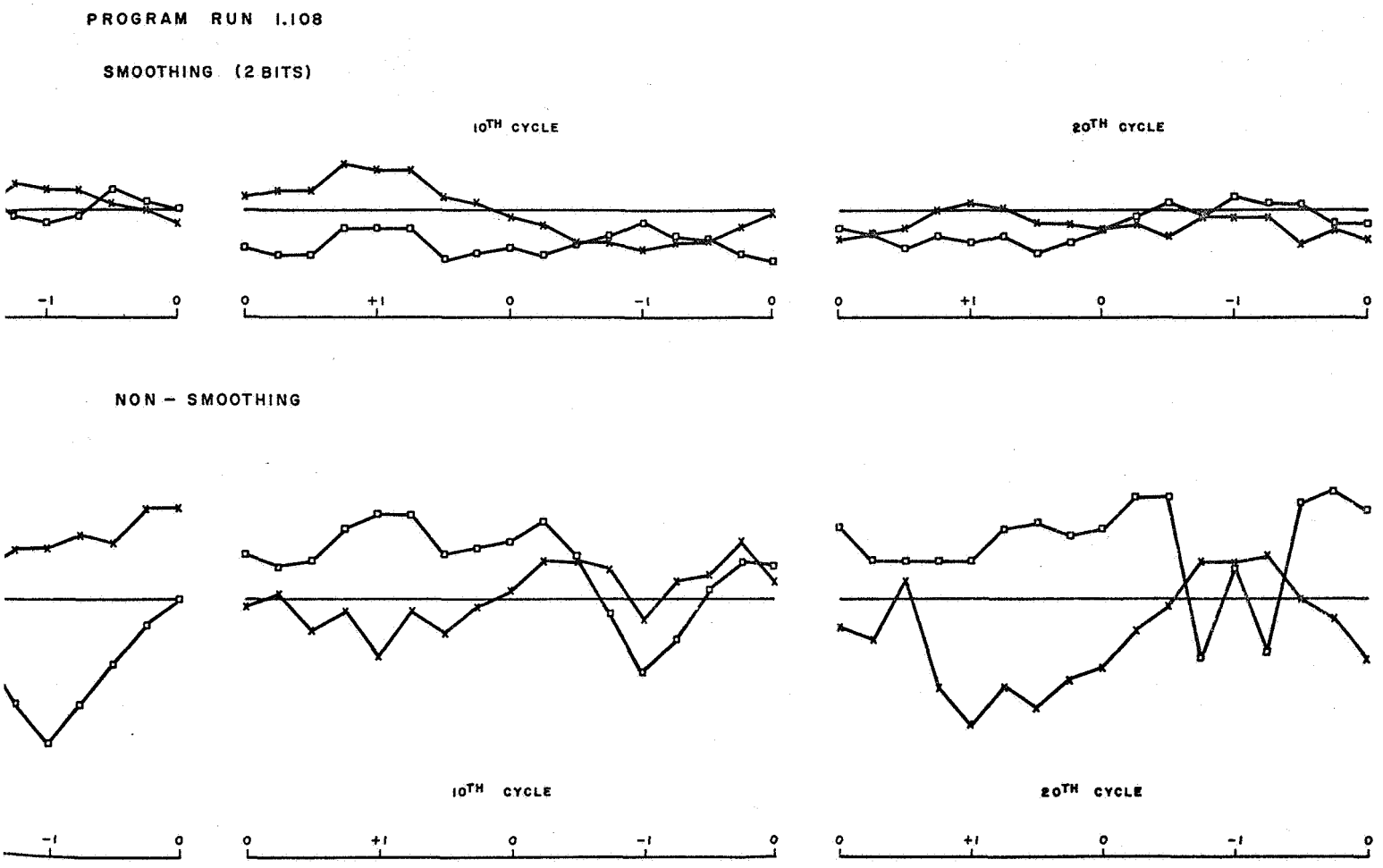
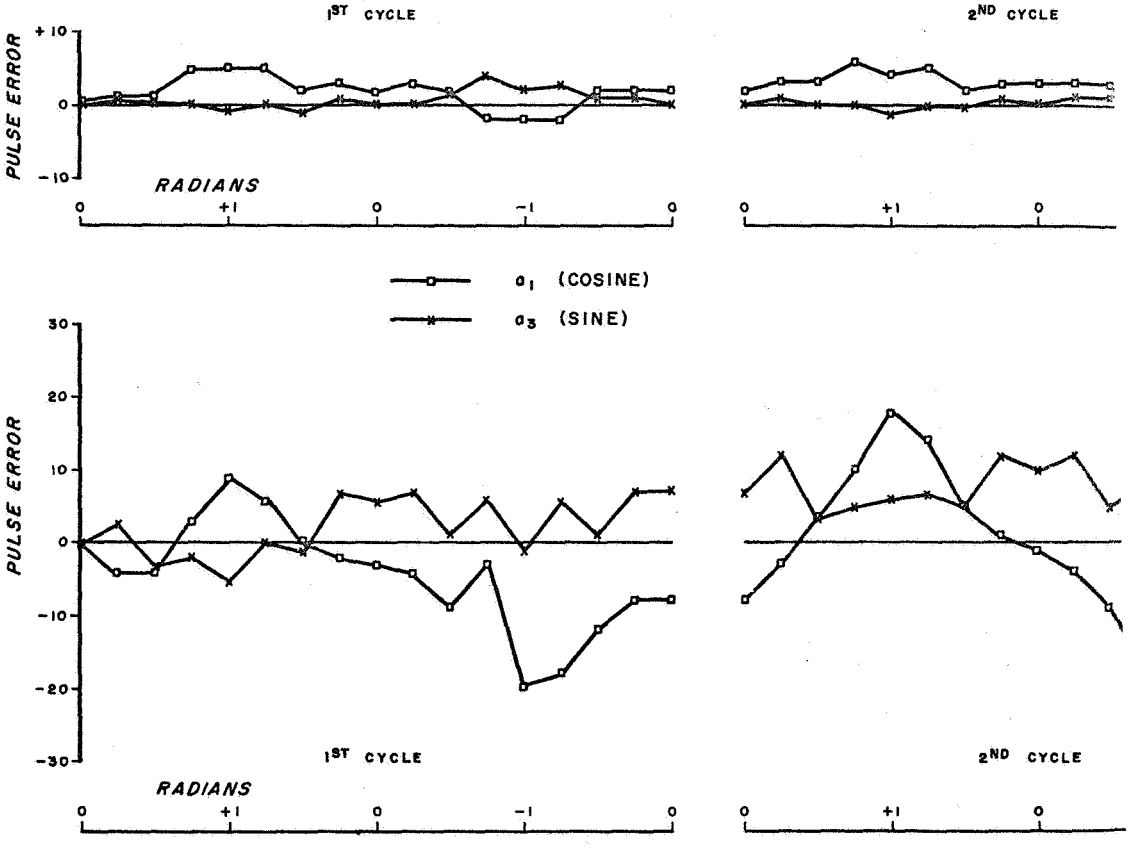


Figure 3.6 Sine-Cosine Error Curve – Smoothing vs. Non-Smoothing


1ST CYCLE
2ND CYCLE
PULSE ERROR
+10
0
-10
RADIANS
0
+1
0
-1
0
a_1 (COSINE)
a_3 (SINE)
30
20
10
0
-10
-20
-30

4.0 ESTIMATE OF AN AIRBORNE COMPUTER – TASK III

4.1 Basis of Estimates

The estimates which follow were prepared for a flyable computer implementing nine direction cosines, operating from asynchronous input pulses representing angular increments. Two bits for smoothing and upscaling at the inputs are included. Outputs provided are pulse increments for inertial velocity components; parallel binary code values for the direction cosines are also available.

All the counters have independent DC reset inputs not used for any other purpose. No specific method of initialization has been included, yet it seems likely that a serial input method using direction-cosine code feedback would be adequate and would involve very little additional circuitry. Some means of memory – in the event of power failure – may be advisable, but has not been provided for in this computer.

It was determined that a 3-slot clock generator would be adequate for the computer, and this has been included. The requirements and characteristics of the associated power supply have also been estimated, and these appear on the graphs below.

4.2 Summary of Results

Figure 4.1 shows the clock performance capability of each of the four computers projected as a function of the angular increment selected. Computers T-1 and T-2 utilize Texas Instruments Series 51 integrated circuits, while Computers F-1 and F-2 use Fairchild Micrologic. The ascending sequence of clock frequency capability is evident.

Figure 4.1 also shows the clock frequency performance requirement as a function of angular increment for various full-scale angular rates. This requirement is determined by the relation:

$$\text{Clock Frequency} \geq \frac{\text{Angular Rate} \times \text{Smoothing Factor} \times \text{No. of Slots}}{\text{Angular Increment}} \qquad (4\text{-}1)$$

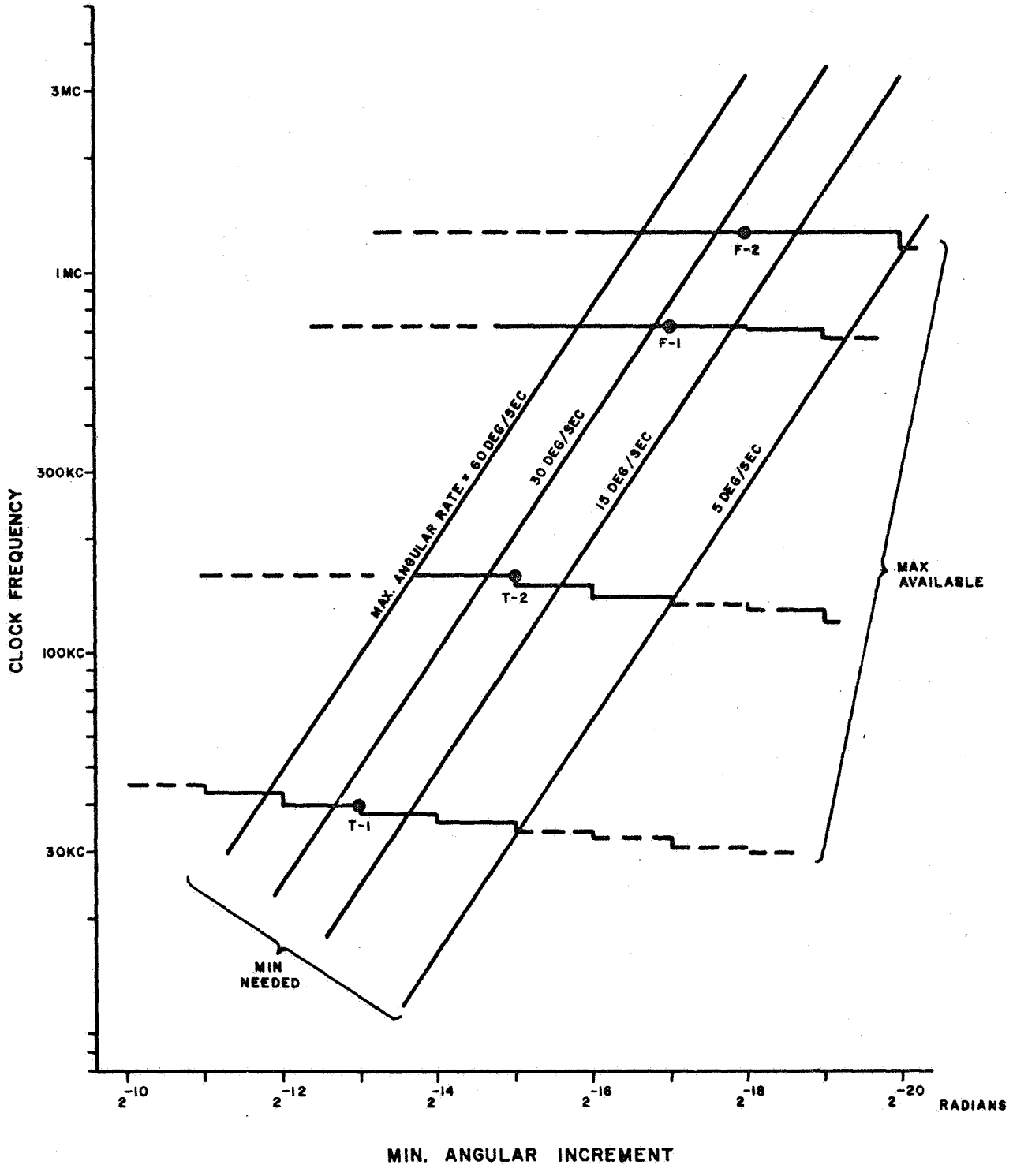


Figure 4.1 Clock Performance Profile

Comparison between the two sets of curves shows that computer T-1 can handle 2^{-13} radian increments at 20 deg/sec, involving a clock frequency of 40 Kc. Similarly, 2^{-15} radian is handled by T-2 at 160 Kc, 2^{-17} radian by F-1 at 720 Kc, and 2^{-18} radian by F-2 at 1.3 Mc.

The component count for these computers is shown in Figure 4.2. About 90% to 95% of the count comprises integrated circuits, so that these figures can be used to obtain failure rate estimates from the appropriate basic data. Computer F-2 differs from F-1 by using standard micrologic instead of Milliwatt Micrologic, and therefore has the same component count.

Figure 4.3 shows the power drain for each computer, both including and excluding power supply. Each computer uses two regulated supply voltages, +3V and +6V, with virtually all the load at the lower voltage. The power drain of the F-2 computer is very large compared to the others.

The weight and volume estimates are shown as Figures 4.4 and 4.5. The F-2 computer was projected using the TO-5 element package, due to power dissipation considerations. The F-1 computer uses the square Fairchild flat package. For these reasons, the weight and volume of the F-2 computer are extremely large compared to the others.

All four computers have been projected in terms of seven to nine types of encapsulated subassemblies of identical size and shape. The quantity of subassembly modules varies from 32 for T-1 up through 80 for F-2. Provisions have been planned for access to and replacement of these modules.

The component count breaks down into six to nine types of integrated circuit elements and three to six generic types of discrete components. The T-1 computer consists of three standard integrated circuit elements (SN514, SN515 and SN518A), three custom elements (SN415-417) and six generic types of discrete components. The T-2 computer has three additional integrated circuit elements in addition to the component types used in the T-1.

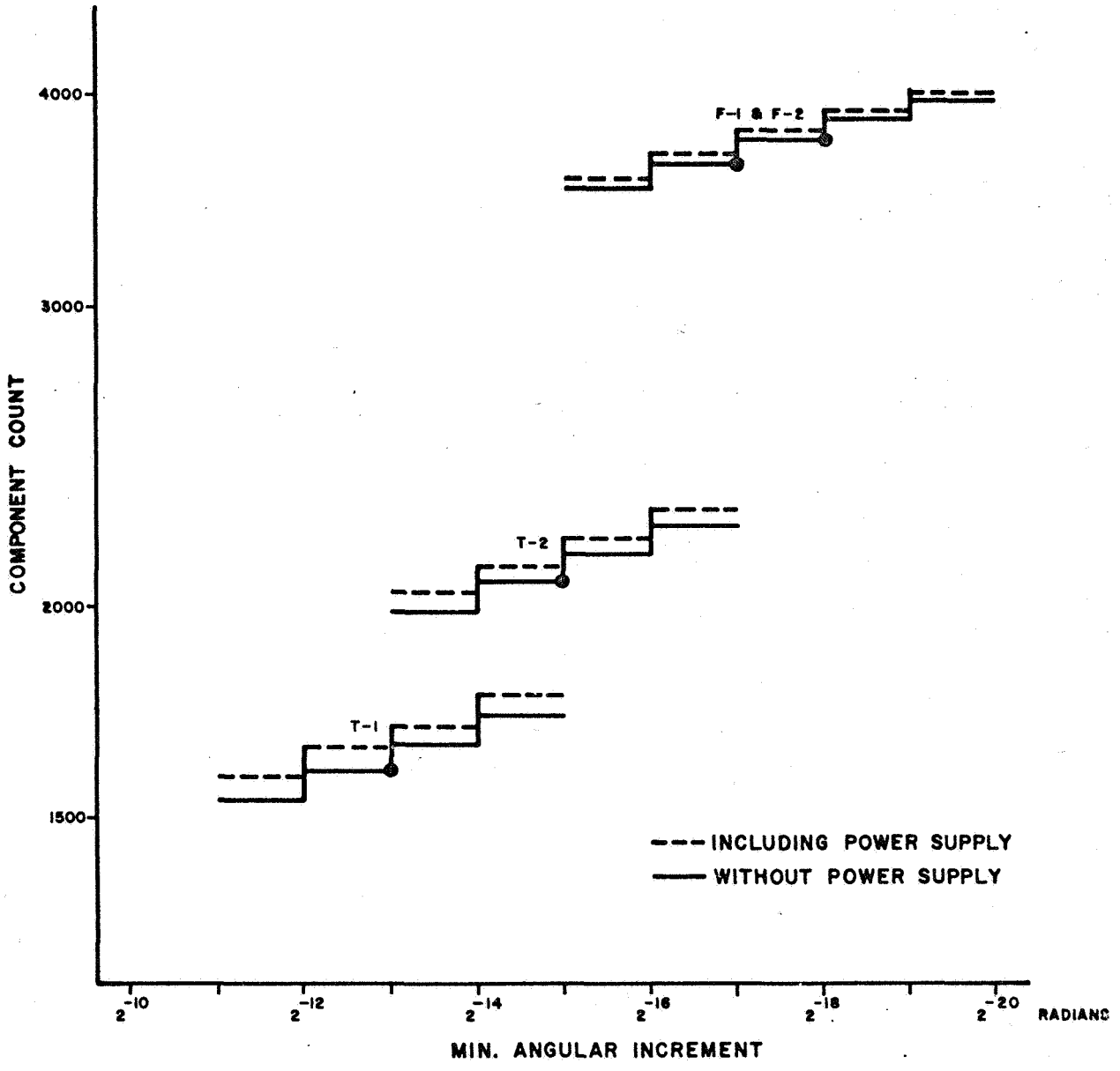


Figure 4.2 Component Count

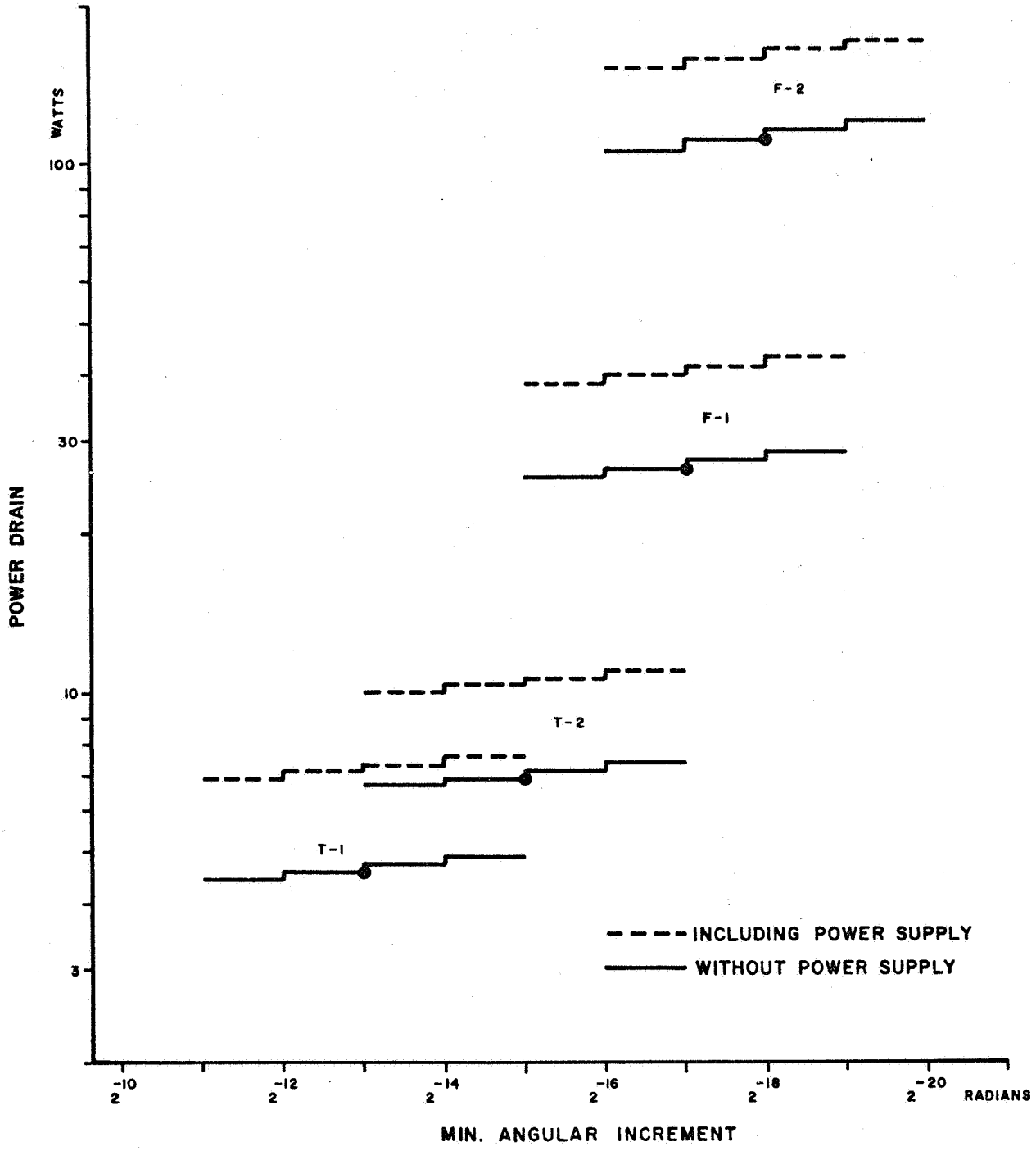


Figure 4.3 Power Drain

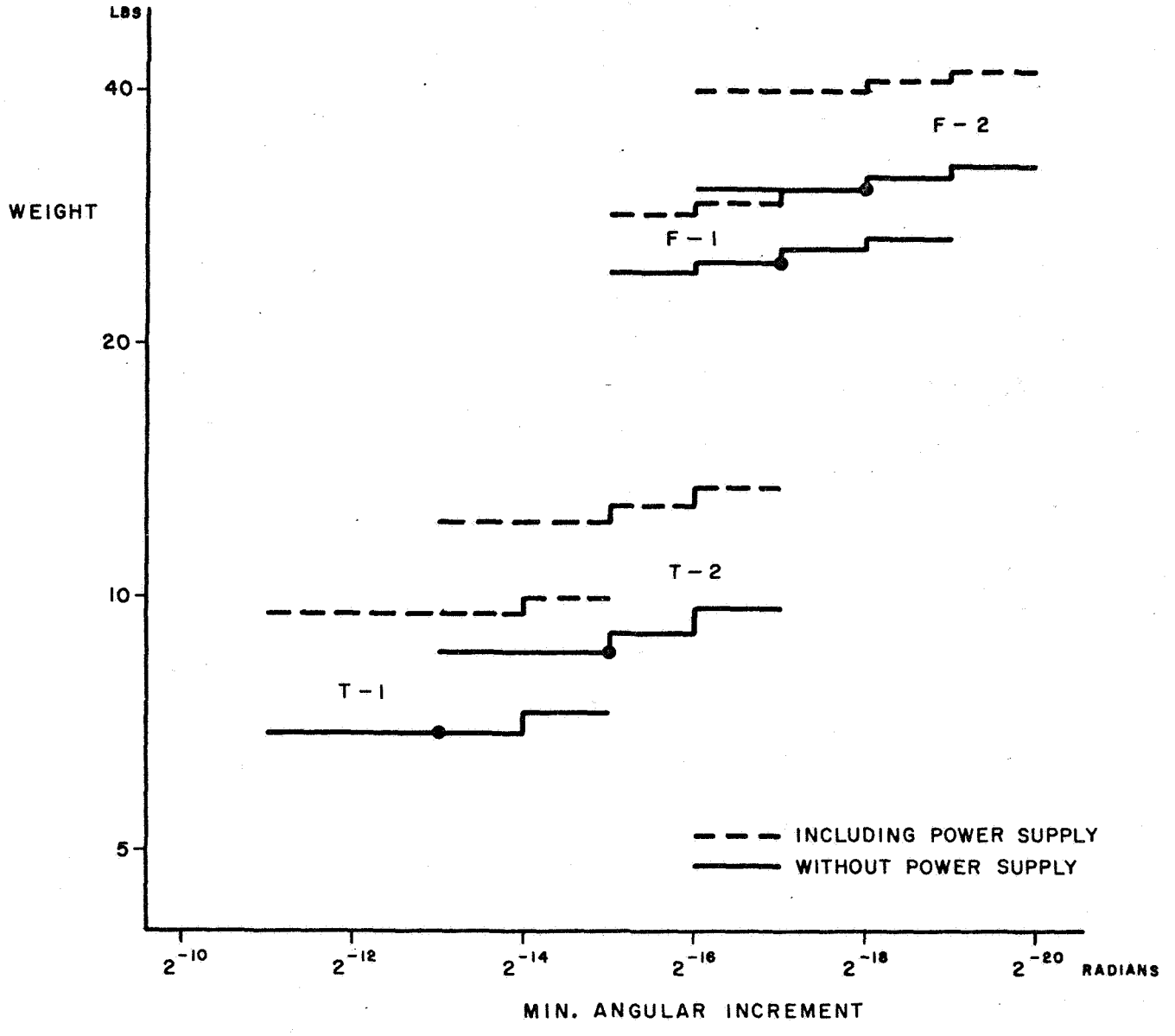


Figure 4.4 Weight

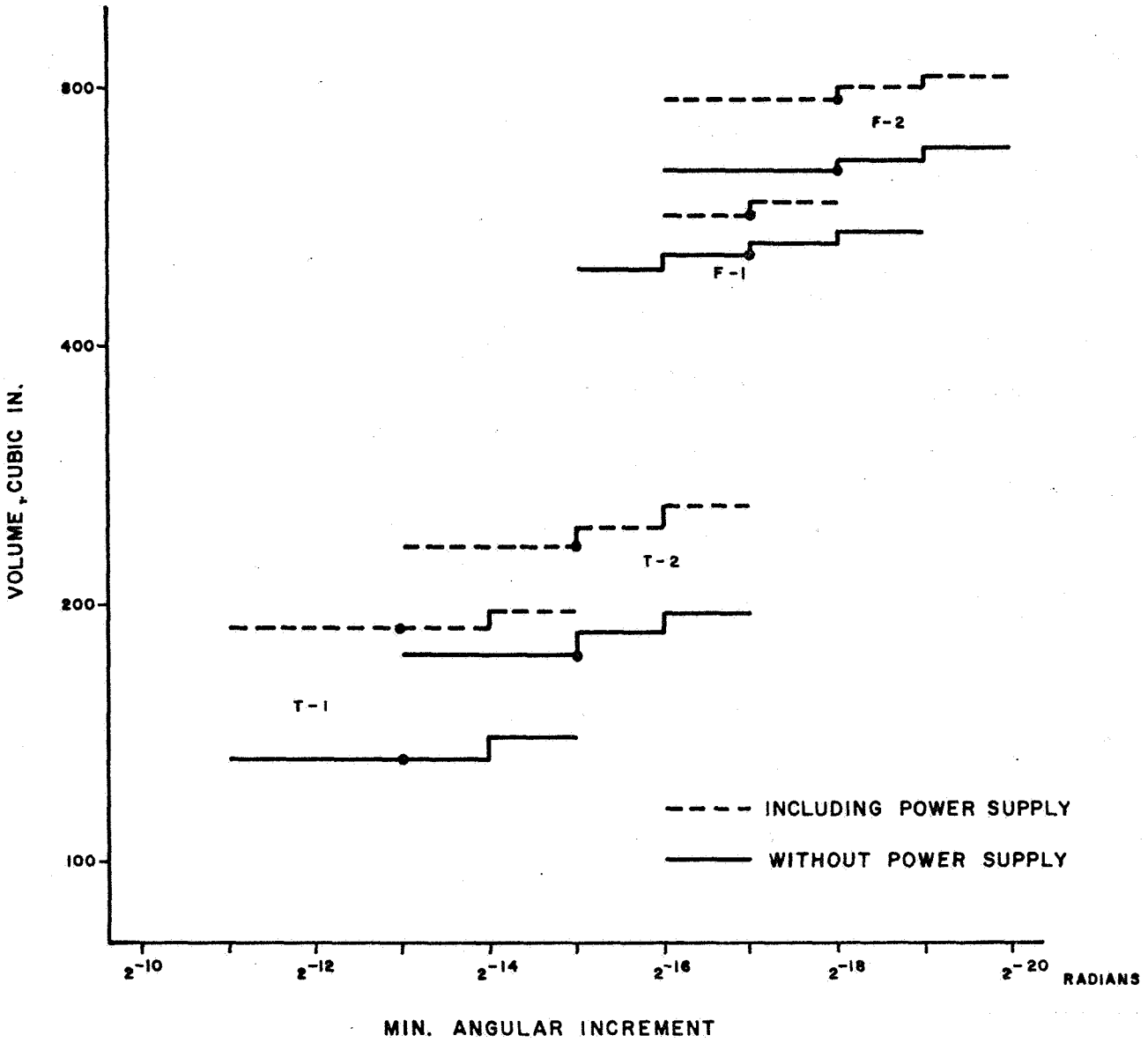


Figure 4.5 Volume

The F-1 computer consists of seven Milliwatt Micrologic elements (900, 909-913-921) and three types of passive discrete components. The F-2 computer comprises nine Micrologic elements (900, 904, 907, 909-11, 913 914 and 916) and three types of passive discrete components. Block diagrams showing the logic arrangement for all four computers appear in Appendix C.

Discrete component usage is confined to the clock slot generator, which includes an astable multivibrator and consists of one encapsulated subassembly.

Power requirements for the T-1 computer are 1.4A at +3V -0, +6% total regulation (line, load, temperature, setup and aging), three ohms maximum source impedance (DC through 2 Mc); and 0.12A at +6V +0, -6% total regulation, 50 ohms maximum source impedance. The T-2 computer requires 2.4A at +3V -0, +6% total regulation, one ohm maximum source impedance; and 0.1A at +6V +0, -6% total regulation, 30 ohms maximum source impedance.

The F-1 computer requires 10A at +3V ±5% total regulation, two ohms maximum source impedance (DC through 10 Mc); and 0.01A at +6V ±5% total regulation, 100 ohms maximum source impedance. The F-2 computer requires 40A at +3V ±8% total regulation, two ohms maximum source impedance (DC through 20 Mc); and 0.01A at +6V, as for the F-1.

The associated power supply projected for use with these computers comprises 46 discrete components, is planned as an adjacent assembly, and operates from +27V ±4V primary power. Its efficiency ranges from 66% with the T-1 load to 71 with the F-2 load.

4.3 Interpretation of Results

The above results define the penalties attached to improved performance in terms of four selected computer configurations. The reasons for their selection

and the underlying technical factors will be examined, in order to provide a background for considering the effects of alternate and future developments in integrated-circuit technology.

Computer T-1 uses the smallest possible quantity of presently-available integrated circuits to implement the desired functions, as far as we know, and therefore offers the best reliability potential. Although clock frequency capability of this computer (see Figure 4.1) is limited, the power drain and weight appear to be the lowest achievable using integrated circuit lines presently in production and generally available. Advantage has been taken of custom interconnections using the Texas Instruments Series 51 Master Slice. The recently-announced Series 53 has higher speed and many more equivalent parts per package, but these potential advantages are partially offset by the lower resistance values and smaller quantity of capacitors.

Computer T-2 provides an improvement in clock frequency capability over Computer T-1 by substitution of look-ahead carry for ripple carry in the forward-backward counters. The added component count, power drain and weight are quite moderate. A substantial additional improvement in clock capability appears feasible by turning to the Series 53, at the cost primarily of a substantially increased power drain. It appears unlikely that any clock frequency improvement over the T-2 can be obtained using the Series 51.

Computer F-1 uses what is believed to be the smallest number of standard Fairchild integrated circuits needed to implement the desired functions. It was decided to avoid custom Fairchild designs because of the extensive mask preparation and production tooling involved; custom design would reduce the component count only about 20 percent. The presently marketed Fairchild flat package is twice as wide and 1.5 times as thick as the Texas Instruments flat package, so a significant volume and weight penalty results. The clock frequency capability, however, is substantially improved over the T-2 computer, in approximate proportion to the power drain increase. The dissipation density (power drain ÷

volume) is comparable with the T-2 computer, due to the use of Milliwatt Micrologic.

The F-2 computer provides nearly twice the clock frequency capability of the F-1 by making use of standard Micrologic. The component count is essentially unchanged, but the guadrupling of the power drain is a major penalty. Dissipation density considerations raise the volume and weight to a point where the flat package is no longer of much advantage over the TO-5 style can.

Other available integrated circuit lines were surveyed briefly to determine whether a better ratio of clock frequency capability to power drain could be obtained. The Sylvania line appears more advantageous than standard Micrologic, and suitable for the same clock frequency range. The other lines available in this frequency range, however, offer poorer performance ratios. For clock frequencies two to three times higher, only the Motorola MECL line appears to be adequate, in terms of presently marketed monolithic silicon integrated circuits. The performance ratio for the Motorola line is intermediate between the Sylvania and standard Micrologic values.

The general speed-power limitation evidenced in present integrated-circuit lines is somewhat amenable to circuit design ingenuity. It can be effectively removed to a significant extent in the future by better substrate isolation techniques which reduce stray capacitance materially, and also by improvements in diffusion techniques that make isolated complementary transistors available on a common substrate.

5.0 CONCLUSIONS

The Error Analysis, Task I, resulted in a definition and mathematical model of the error sources and error propagation of the attitude computation problem. It was determined that only three error sources need be considered; namely: truncation, integrand random quantization, and integrand sign-reversal errors. A special error-propagation formula was derived as it was found that the general method, as described in the Appendix, was not applicable. This is due to the fact that the error sources were larger than the error build-up through error propagation in the closed-loop attitude computation. It was also determined that the integrand biased quantization source error can be neglected since, in the negative-feedback closed-loop program, the error will not grow to more than 2 pulses. The error-propagation analytical model was derived independently of the error-source equations. Upon application of the error-propagation equations alone it was found that pseudo-sinewave curves were generated with undetermined peak amplitudes. In order to ascertain the values of the peak amplitudes, the error-source equations were utilized. It was impossible to predict the bias of the error curves due to the lack of knowledge of the boundary conditions of the differential equations which describe the error curves.

The error analysis is more applicable in those cases where the dynamic range of the inputs (number of pulses of the independent variable $\Delta\theta_i$) is several orders of magnitude greater than the number of sign reversals of the integrand.

The predicted errors versus the experimental results as found in MEDOC Simulation, Task II, demonstrated fairly good agreement in those cases which were investigated. Figures 5.1 through 5.3 illustrate the expected and worst-case values of the errors within 20 revolutions/cycles as a function of 2^{-11} to 2^{-20} resolution angular inputs. The reader will note, after reviewing the figures, that the experimental values were more nearly those of the predicted expected value than the predicted worst case. Since the experimental investigation was only concerned with 2^{-15} radian input resolution and the number of programs

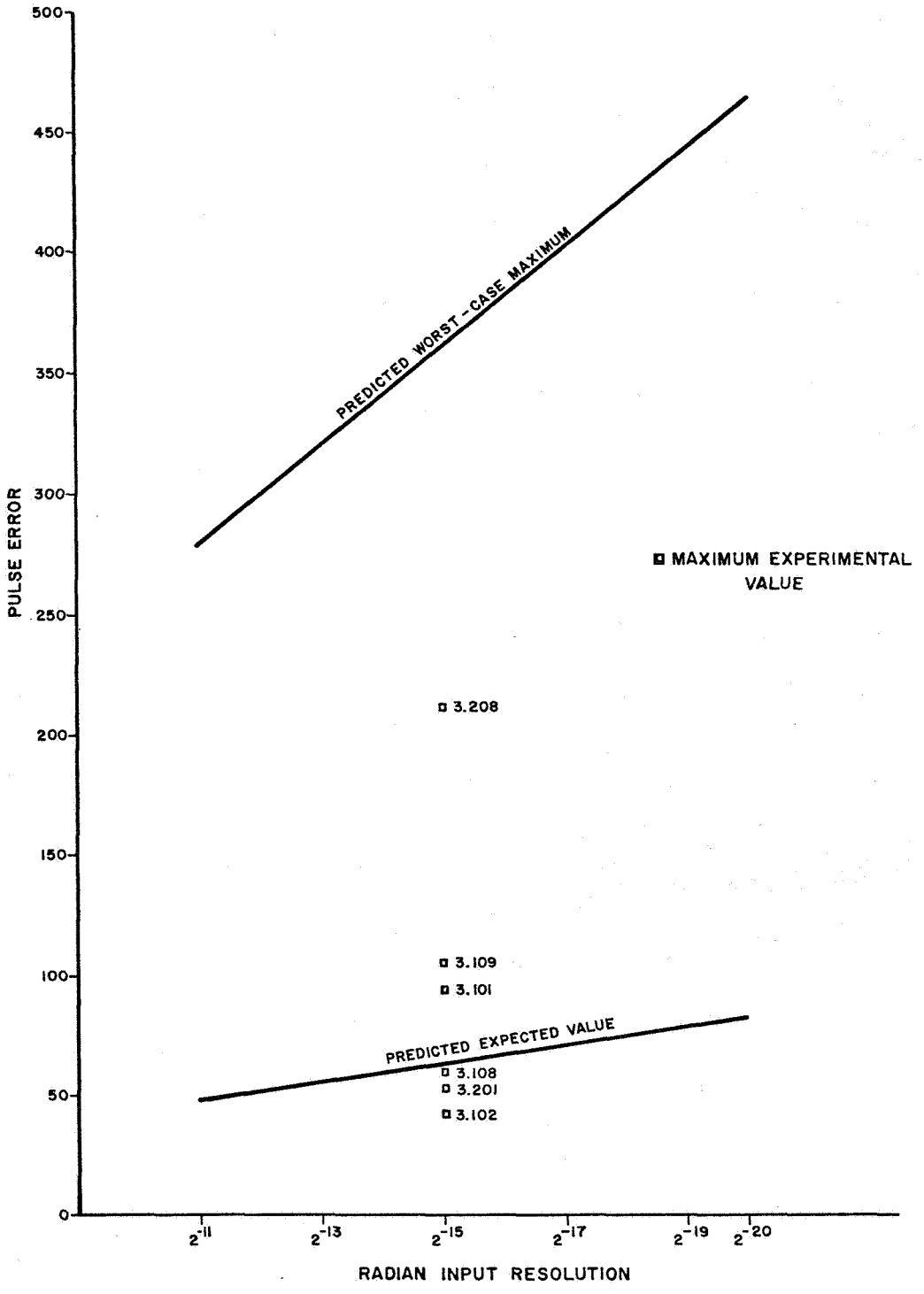


Figure 5.1. Expected and Worst-Case Errors, Predicted and Experimental (Rotation)

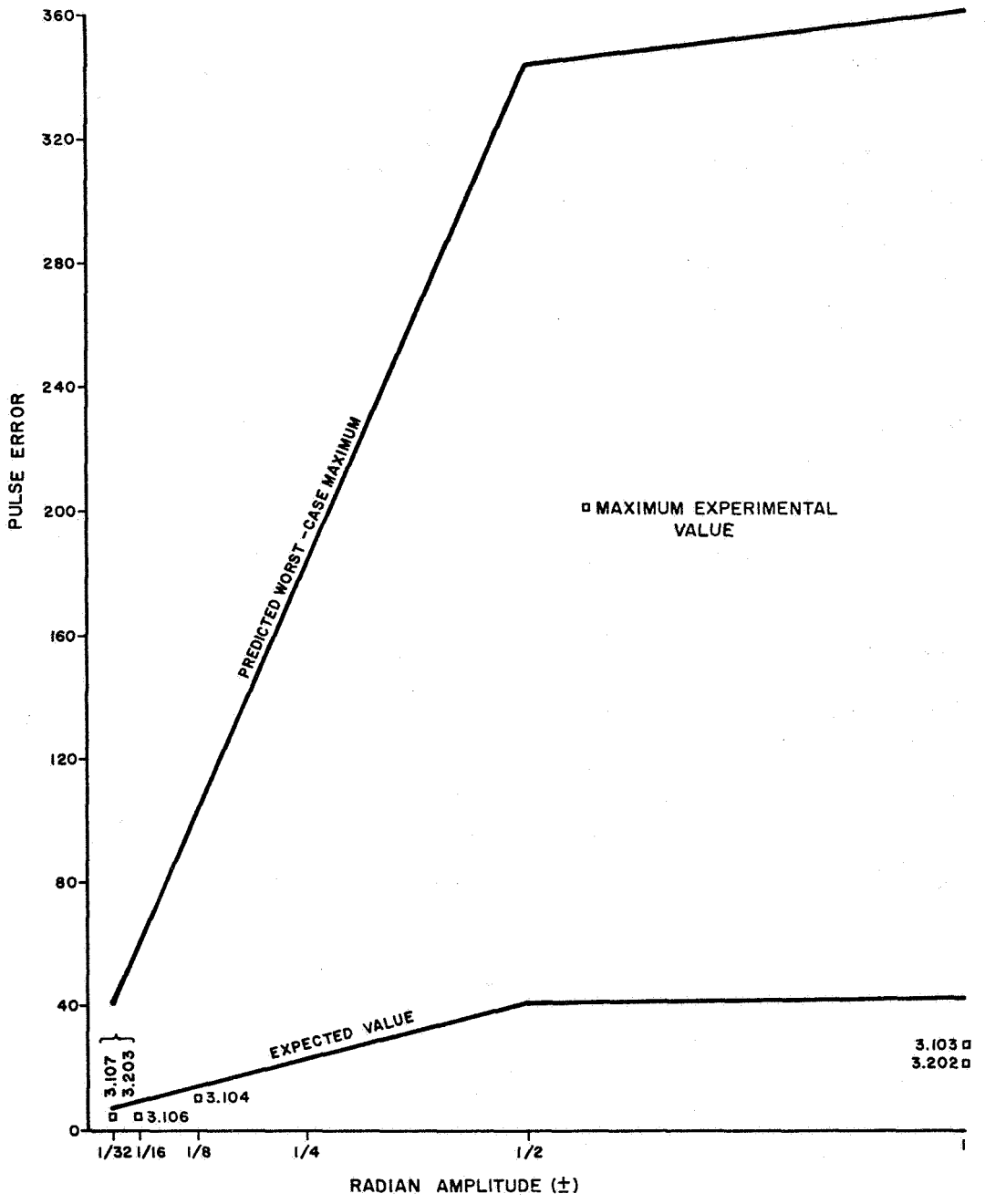


Figure 5-2. Expected and Worst-Case Errors, Predicted and Experimental (Limit Cycle)

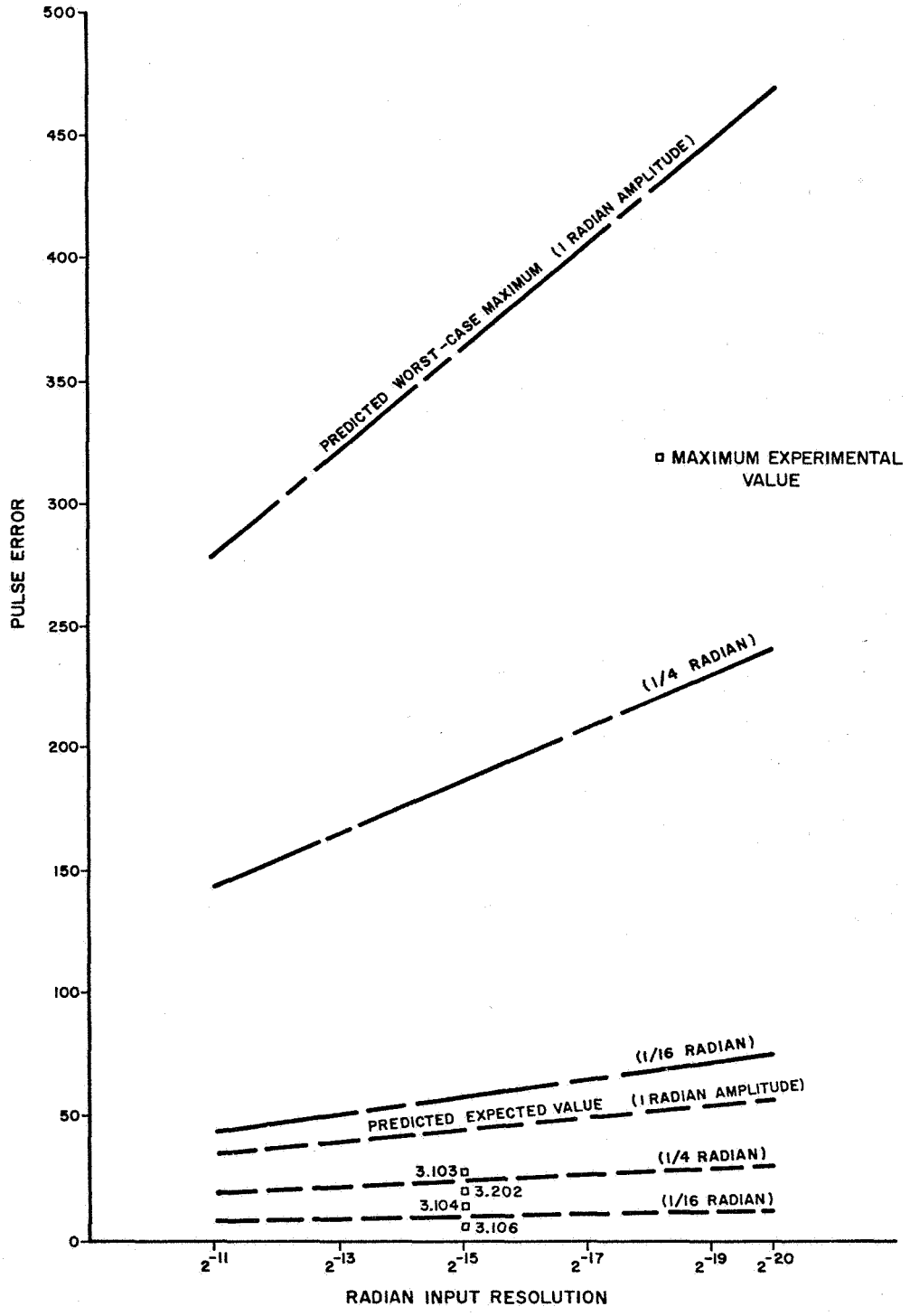


Figure 5.3. Expected and Worst-Case Errors, Predicted and Experimental (Limit Cycle $\Delta\theta_i = 2^{-15}$)

run were limited, the "closeness of fit" of the mathematical models is difficult to assess. Obviously, more work in this area is required. The importance of smoothing was demonstrated early in Task II and all results and conclusions as presented herein, include smoothing.

Error analysis of the velocity equations did not appear to be warranted as the experimental results demonstrated negligible errors in regard to the programs run. Since a simple open-loop program consisting of multiplication and addition was involved, it was decided to spend most of the time investigating the direction-cosine computation. The results of the velocity programs, as illustrated in Appendix B, are exceptionally good inasmuch as a maximum pulse error of 3 pulses was observed. This indicates a maximum error of 0.3 ft/sec. after a total change in body velocity of $\Delta V_x^B = 2150$ ft/sec.

Estimates of an airborne computer resulted in four basic configurations; two each utilizing the TI and Fairchild micrologic units. Today's state-of-the-art torque gyro resolution is in the order of 2^{-14} radians, and can be pulsed to sense 30-degree-per-second body rates. Thus, the T-2 computer, as configured in Section 4, is applicable to the present rate gyro inputs. This configuration implies a:

Component Count:	2000
Power Drain:	7 watts
Weight:	8.5 pounds
Volume:	180 cu.in.

Predicted Expected Error after 20 Revolutions: 60 pulses

Predicted Expected Error after 20 Limit Cycles (1 radian): 42 pulses

(1/4 radian): 23 pulses

(1/16 radian): 8 pulses

The hardware specification tradeoffs are easily determined for any application from 2^{-11} to 2^{-20} radian input resolutions and up to 60° per second body rates by reference to Figures 4.1 through 4.5. In addition, Figures 5.1 and

5.2 can be utilized to predict the expected and worst-case errors within 20 revolutions/cycles. Also available are the mathematical models that can be utilized for error prediction of specific problem investigations.

Although not part of the work statement, an attempt was made to simulate a DDA rectangular integration algorithm on the IBM 1401 to compare the results with the MEDOC Runs. Due to time limitations and programming difficultites in terms of sequencing the arithmetic operations, the attempt proved to be unsuccessful. Although no conclusions can be drawn, two results are worth noting. First, program run 2.101 comparison of simulated DDA versus MEDOC resulted in a 76 maximum DDA pulse error versus a 27 maximum MEDOC pulse error. On the other hand, program run 1.101, as programmed on a TRICE DDA, resulted in a 33 maximum TRICE DDA pulse error versus a 47 maximum MEDOC pulse error. Obviously, more work has to be accomplished in the DDA area in order to arrive at realistic conclusions regarding the comparison of DDA versus DO. Nevertheless, LFE believes intuitively that, in terms of an error analysis, both the DDA (Rectangular Integration) and DO techniques are in the same "ball park". It is suggested, therefore, that the advantage is in favor of the DO techniques due to the potential 25% reduction in hardware characteristics such as component count, power, weight, and volume. Final determination can be achieved by performing the same work tasks in a separate DDA study.

The significance of this report is the fact that a DO special-purpose computer can be specified for the attitude computation of a strapped-down guidance system. Specifications in terms of clock frequency, component count, power, weight, volume, and predicted expected errors can be written knowing the input requirements such as angular and velocity resolutions, and body rates. Application of Figures 4.1 through 4.5 and Figures 5.1 and 5.2 allow the reader the facility of specifying a DO Computer for any attitude computation application. Recommendations for further work in this area are outlined in Section 6.

6.0 RECOMMENDATIONS

The study of the Digital-Operational Technique demonstrated its applicability to the attitude computation of a strapped-down guidance system. However, in the course of the work it became apparent that several areas require further investigation. Recommendations relating to these areas are presented below.

6.1 Study Areas

It is suggested that further study be conducted to determine the "closeness of fit" of the mathematical models and experimental results. Included in this study should be the investigation of reversibility of very small limit cycle amplitudes over an extended number of cycles where the number of sign reversals is the same order of magnitude as the number of input angular pulses. It is this type of program that difficulty in applying mathematical models is expected. Either new models or a pulse-by-pulse analysis may have to be utilized. Clock sequencing should also be investigated in an attempt to reduce the MEDOC errors as well as other methods of smoothing without the need of upscaling the input data rate.

Comparison with other special-purpose computer techniques, such as DDA, should be made based upon identical work tasks as applied in the DO study.

6.2 Hardware Areas

It is suggested that both a breadboard and prototype DO attitude computer be built to demonstrate the feasibility of the packaging concepts as discussed in this report. The T-2 computer is recommended as it can be used with present state-of-the-art gyros and will provide a valuable simulation tool in the laboratory.

APPENDIX A

A.1 DO PULSE MULTIPLIER ANALYSIS: CASE OF CONSTANT Y-COUNTER

A.1.1 The DO pulse-multiplier circuit, sometimes called BRM has been described in the literature (cf. bibliography) and is illustrated in Figure A.1. An exact formula for the output number of pulses can be established in the case of a constant integrand Y = A.

In order to derive an analytical expression of the number of pulses at the output of the multiplier, the number of X input pulses is best described by the polynomial

$$X \equiv \sum_{j=0}^{j=n-1} x_j 2^j + N2^n \quad , \tag{A-1}$$

where N is an integer ≥ 0, and where the xj's can take on the value zero (0) or one (1). The constant integrand, input A, to the multiplier is best described by its own binary code

$$A \equiv \sum_{i=0}^{i=n-1} a_i 2^i \quad , \tag{A-2}$$

where any $a_i = 1$ means that the corresponding AND gate of the multiplier is open to let the transition pulses of the $(n-i)^{th}$ order counter stage of X go to the multiplier output OR gate. It can be easily shown that this number of pulses is given by the following expression for the output of the AND gate at a_i;

$$\left(x_{n-i-1}\right) 2^0 + \sum_{j=n-i}^{j=i-1} x_j 2^{j-1} + N2^{n-1} \tag{A-3}$$

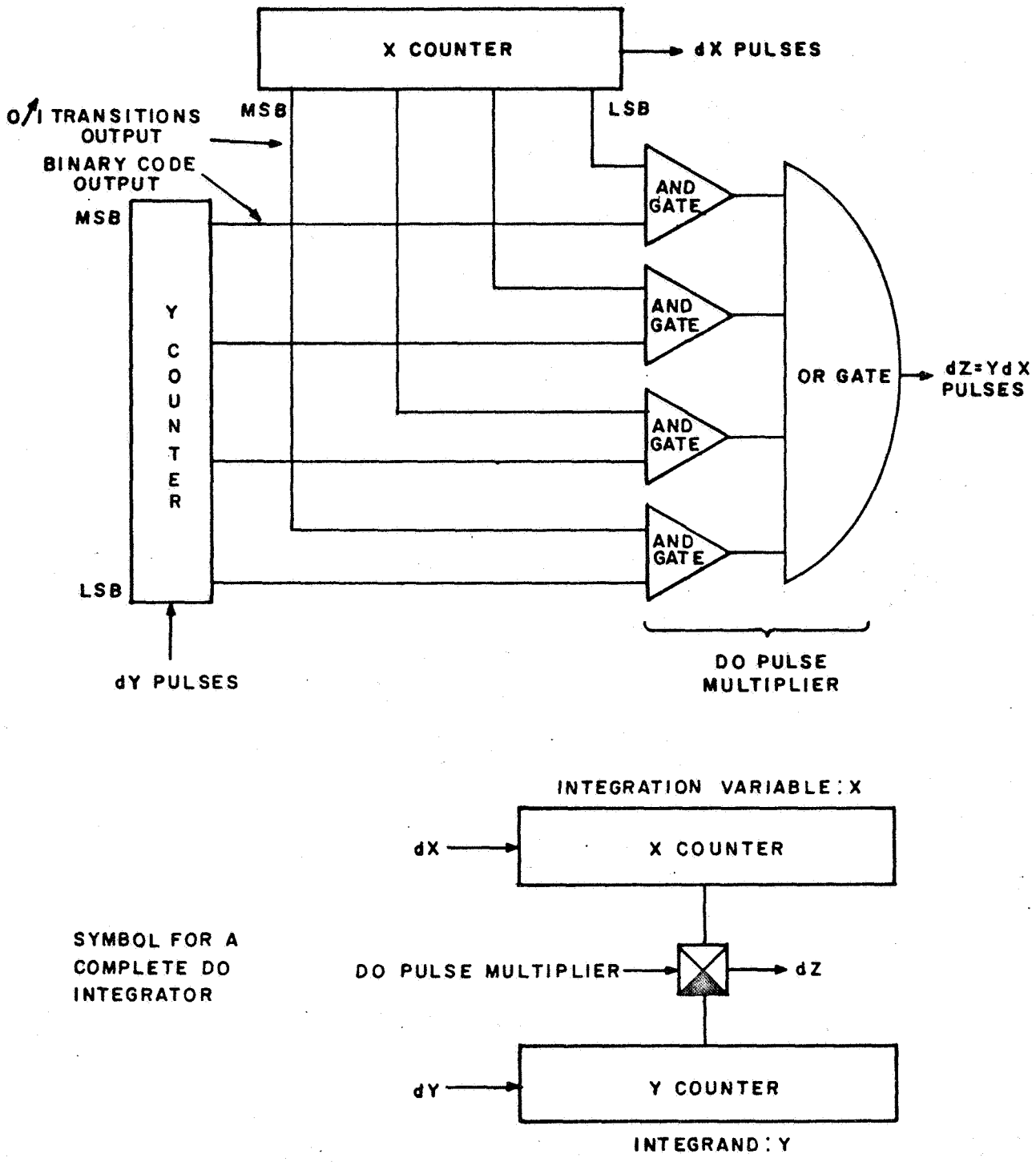


Figure A.1. DO Pulse Multiplier Logic Diagram

The summation of the pulses at the output of every AND gate $a_i = 1$ will give the analytical expression of the number of pulses at the output of the multiplier. Table A-1 relates the determination of the number of output pulses when adding the values found at the crosspoints of the selected a_i's and x_j's.

The absolute error at the multiplier output is the difference between the number of output pulses and the product $A \cdot X \cdot 2^{-n}$. It is expressed analytically through use of identities A-1, 2, and 3 as in Table A-2 which can be used in the same way as Table A-1. An expression for the maximum positive absolute error has already been developed, Reference 20. Use of the series summation method shows that this maximum positive error occurs after three quarters of full scale of both A and X, and has the value

$$\frac{n}{6} + \frac{17}{18} + (-1)^n \frac{2^{-n}}{9} = \epsilon \qquad \text{(A-4)}$$

A similar series summation method shows that the maximum negative error happens in the vicinity of one third of full scale of both A and X. This maximum negative error has the value:

$$\frac{n}{6} + \frac{8}{9} + (-1)^n \frac{2^{-n}}{9} = \epsilon \qquad \text{(A-5)}$$

The maximum positive error is always larger than the maximum negative error by the amount $\frac{1}{18}$ pulse value. In particular, there is no error when the independent variable reaches full scale.

Table A-1 can be used to derive an analytical exact expression of the number of output pulses for a typical function generation, where the integrand varies as a simple function of the independent variable (power function, exponential, sine, cosine, generator, etc). Two situations may occur during generation:

TABLE A-1

PULSE MULTIPLIER OUTPUT

X \ A	a_0	a_1	a_2	a_3		a_{n-3}	a_{n-2}	a_{n-1}
X_0	0	0	0	0		0	0	1
X_1	0	0	0	0		0	1	1
X_2	0	0	0	0		1	1	2
X_3	0	0	0	0		1	2	4
X_{n-3}	0	0	1	1		2^{n-6}	2^{n-5}	2^{n-4}
X_{n-2}	0	1	1	2		2^{n-5}	2^{n-4}	2^{n-3}
X_{n-1}	1	1	2	4		2^{n-4}	2^{n-3}	2^{n-2}
N	n	2N	4N	8N		2^{n-3}(N)	2^{n-2}(N)	2^{n-1}(N)
TOTAL								

(1) The independent variable advances one or several pulses for one increment of the integrand. Table A-1 then gives the number of output pulses:

$$P = \text{Total}\left\{A_i(X + \Delta X)\right\} - \text{Total}\left\{A_i X\right\} \qquad \text{(A-6)}$$

where A_i is the initial integrand.

(2) The integrand advances several increments (n) for one advance of the independent variable. Table A-1 gives the number of output pulses:

TABLE A-2

PULSE MULTIPLIER ERROR OUTPUT

X \ A	a_0	a_1	a_2	a_3		a_{n-3}	a_{n-2}	a_{n-1}
X_0	$-\frac{1}{2^n}$	$-\frac{1}{2^{n-1}}$	$-\frac{1}{2^{n-2}}$	$-\frac{1}{2^{n-3}}$		$-\frac{1}{8}$	$-\frac{1}{4}$	$+\frac{1}{2}$
X_1	$-\frac{1}{2^{n-1}}$	$-\frac{1}{2^{n-2}}$	$-\frac{1}{2^{n-3}}$			$-\frac{1}{4}$	$+\frac{1}{2}$	0
X_2	$-\frac{1}{2^{n-2}}$	$-\frac{1}{2^{n-3}}$	$-\frac{1}{2^{n-4}}$			$+\frac{1}{2}$	0	0
X_3						0	0	0
-	-	-	-	-	-	-	-	-
X_{n-3}	$-\frac{1}{8}$	$-\frac{1}{4}$	$+\frac{1}{2}$			0	0	0
X_{n-2}	$-\frac{1}{4}$	$+\frac{1}{2}$				0	0	0
X_{n-1}	$+\frac{1}{2}$					0	0	0
N	0	0	0	0		0	0	0
TOTAL								

$$P = \sum_{p=o}^{p=n} \left[\text{Total } \left\{ A_{i+P+1} \ (X_i) \right\} - \text{Total } \left\{ A_{i+P} \ (X_i) \right\} \right. \tag{A-7}$$

the calculation will involve only summation of truncated series of the geometric type.

However, the general case will not be handled easily with these formulae since the integrand driving function is not usally known exactly, pulse-by-pulse. Instead, typical worst-case error build up will be analyzed pulse-by-pulse, and the general case will be compared to those worst cases with the introduction of form factor and statistical reductions.

A.2 DO WORST-CASE ERROR FORMULAE: CASE OF A VARYING Y-COUNTER

A.2.1 Maximum Truncation Error in the Number of Output Pulses From a Pulse Multiplier When the Y-Counter Varies During the X-Counter Variable Incrementation

The following analysis is purely deterministic and the worst case will always be considered, whatever may be its low probability of occurrence. It has been shown in the literature, Reference 21, that if the integrand oscillates around the 1/2 full scale value, it is possible to get no output pulses. We shall use the equivalent statement.

A.2.1.1 Rule: Every time the integrand reverses its direction of incrementation, it can result in one output erroneous pulse. For the rest of the time, we shall deal with monotonic driving functions of the integrand.

A.2.1.2 Maximum Negative Error for Full-Scale-Change of the Integrand When Y Can Vary Faster Than X

Such a maximum error will be reached if the integrand goes to a maximum value after remaining as large as possible, without generating one output pulse, during the incrementation of the integration variable. Table A-3 illustrates this worst case for a 4-bit multiplier. The analytical expression of the maximum error is derived from the following procedure.

Starting from the bottom line of the Y-counter with a value 1011...1, the Y-counter keeps its value until it reaches an X-counter transition gated with a bit one of the Y-counter. At this point, this bit becomes a zero and the bits to the right (LSB's) become ones, in order to insure no output pulse, yet a maximum Y-counter value.

$$\frac{1}{2^n}\left[(2^{n-1}-2)+2\,(2^{n-2}-2)\ldots+2^{p-1}(2^{n-p}-2)\ldots+2^{n-3}\;\;2^{n-(n-2)}-2\right.$$

$$+2^{n-1}+2^{n-2}(2^2-1)\ldots+2^{n-p}\;\;2^p-1-(2^{p-2}-1)\;\;\ldots$$

$$\left.+2^{n-(n-1)}\;\;2^{n-1}-1-(2^{n-3}-1)\;\;+1\;\;2^{n-1}-1-(2^{n-2}-1)\right]$$

= maximum negative error in pulse value. After rearranging the terms, it becomes

$$= \frac{1}{2^n} \left\{ 2^{n-1}(n-2) - 2^{n-2}\left(1 + \frac{1}{2} + \ldots + \frac{1}{2^{n-3}}\right) + 2^{n-1} + 2^n(n-2) - 2^{n-2}(n-2) + 2^{n-1} - 2^{n-2} \right\}$$

and, after simplification

$$\text{Maximum negative error} = \frac{5}{4}\, n - \frac{9}{4} + 2^{-n+1} \text{ pulse value.} \qquad \text{(A-8)}$$

For example: n = 4; error = 2.8 pulse value

n = 10; error = 10.25 pulse value

It can be seen that for the usual counter size, the maximum negative error is equal to the number of bits of the multiplier.

We stated before that, at full scale, the error would be $1 - 2^{-n}$, and, if the independent variable cycles to full scale several times, no additional error will occur since the integrand is already at full scale and will not vary any further.

A.2.1.3 Maximum Positive Error When Y Can Vary Faster Than X

The integrand remains small, but catches the independent variable transitions. The procedure is as follows; the least significant bit of the integrand goes from zero to one, just before the transition of the MSB of the independent variable. It goes to zero again when the next MSB of the integrand changes from zero to one at the corresponding transition of the independent variable. Table A-4 illustrates this worst integrand progression in a 4-bit multiplier. The analytical expression of this worst positive error for a n-bit multiplier is as follows.

$$\frac{1}{2^n} \left\{ n \times 2^n - (2^{n-2} + 2^1 \times 2^{n-3} \ldots + 2^p \times 2^{n-(p+2)} \ldots + 2^{n-1} \times 2^{n-n}) \right\}$$

and, after simplification,

$$\text{Maximum positive error} = \frac{3}{4}\, n - \frac{1}{4} \text{ pulse value.} \qquad \text{(A-9)}$$

TABLE A-3

MAXIMUM NEGATIVE ERROR FOR
FULL-SCALE CHANGE OF THE INTEGRAND

Advances (Iterations) of the X-Counter	Transitions* of the X-Counter LSB#	Y-Counter MSB 1	2	3	LSB 4
1	1	0*	0	0	1
2	2	0	0*	0	1
3	1	0*	0	0	1
4	3	0	0	0*	1
5	1	0*	0	1	0
6	2	0	0*	1	0
7	1	0*	0	1	0
8	4	0	0	1	0*
9	1	0*	0	1	1
10	2	0	0*	1	1
11	1	0*	1	0	1
12	3	0	1	0*	1
13	1	0*	1	1	1
14	2	1	0*	1	1
15	1				

TABLE A-4

MAXIMUM POSITIVE ERROR FOR FULL-SCALE CHANGE OF THE INTEGRAND

Advances (Iterations) of the X-Counter	Transitions* of the X-Counter LSB#	Y-Counter MSB 1	2	3	LSB 4
1	1	0*	0	0	0
2	2	0	0*	0	0
3	1	0*	0	0	0
4	3	0	0	0*	0
5	1	0*	0	0	0
6	2	0	0*	0	0
7	1	0*	0	0	0
8	4	0	0	0	1* 1 pulse out
9	1	0*	0	0	1
10	2	0	0*	0	1
11	1	0*	0	0	1
12	3	0	0	1*	0 1 pulse out
13	1	0*	0	1	0
14	2	0	1*	0	0 1 pulse out
15	1	1*	0	0	0 1 pulse out

It can be seen that, for the usual size of a multiplier, the maximum positive error is equal to 3/4 the number of bits of the multiplier. No additional error of this type will be encountered at the next cycle to full scale of the independent variable. It must be noted that both dX and dY sequences were positive, showing that the sign of the output error cannot be predicted in the general case.

A.2.2 Example Of The Square Generator Where Y and X Vary At The Same Rate

This example is particularly interesting since it lends itself to an easy pulse by pulse analysis. Small or large error outputs can result depending upon circuit changes such as rate to code pulse sequence and odd or even integration of multiplier bits.

The law of formation of the output pulses can be easily deduced from a few examples with short counter length. Figure A.2 shows the corresponding DO program where the total field of the X-counter output is used as code and rate inputs to the pulse multiplier.

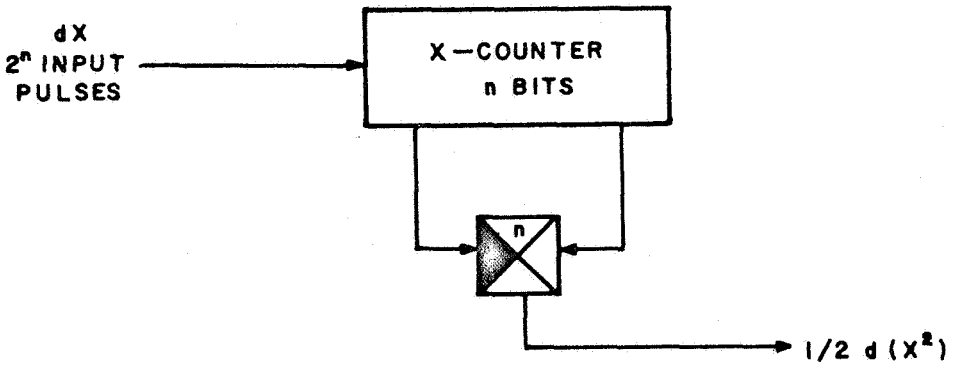


Figure A.2. DO Square Generator

When the X counter is driven monotonically to full scale (2^n advance pulses), the following will be observed.

(a) For n even and code change occurring before transition output

The multiplier output consists of the sum of 1/2 the transitions of the $\frac{n}{2}$ l.s.b. X-counter outputs

$$Z = 2^{n-2} + 2^{n-3} + \ldots . 2^{n-(1+\frac{n}{2})} = 2^{n-1} - 2^{\frac{n}{2}-1}$$

it corresponds to an output error

$$\xi = -2^{\frac{n}{2}-1} \qquad \text{(A-10)}$$

(b) For n even and code change occurring after transition output

The multiplier output consists of the sum of: 1/2 the transitions of the $\frac{n}{2}$ l.s.b. X-counter outputs plus all the transitions of the $\frac{n}{2}$ m.s.b. X-counter outputs

$$Z = 2^{n-2} + 2^{n-3} \ldots . + 2^{n-(1+n/2)} + 2^{n-(1+n/2)} + 2^{n-(2+n/2)}$$

$$+ 2^{n-(n/2+n/2)} = 2^{n-1} + 2^{n-(1+n/2)} - 2^{\circ}$$

it corresponds to an output error

$$\xi = + 2^{n/2-1} - 1 \text{ pulses} \qquad \text{(A-11)}$$

(c) For n odd and code change occurring before transition output

The multiplier output consists of 1/2 the transition of the $\frac{n-1}{2}$ l.s.b. X counter output plus all the transitions of the $(\frac{n+1}{2})$ l.s.b. X-counter output (middle stage)

$$Z = 2^{n-2} + 2^{n-3} + \ldots . 2^{[n-(\frac{n-1}{2}+1)]} + 2^{\frac{n+1}{2}-1} = 2^{n-1}$$

it corresponds to an output error

$$\xi = 0 \text{ pulse} \qquad \text{(A-12)}$$

(d) For n odd and code change occurring after transition output.

The multiplier output consists of 1/2 the transitions of the $\frac{n-1}{2}$ l.s.b. X-counter outputs plus all the transitions of the $\frac{n-1}{2}$ m.s.b. X-counter outputs

$$Z = 2^{n-2} + 2^{n-3} \ldots + 2^{\left[n - \left(\frac{n-1}{2} + 1\right) \right]} + 2^{\frac{n+1}{2} - 2} + 2^{\frac{n+1}{2} - 3}$$

$$+ \ldots + 2^{\circ} = 2^{n-1} - 1$$

it corresponds to an output error

$$\xi = -1 \text{ pulse} \qquad \text{(A-13)}$$

Important conclusions can be drawn from this example. When X and Y vary at the same rate and the multiplier has an even number of stages, the error amplitude will reach one-half the square root of the number of input pulses and its sign will depend upon the rate-out to code-out sequence (i.e., the actual DO circuitry and clocking in use). If the multiplier has an odd number of stages, the error will remain smaller or equal to one pulse. In practice, DO multiplications with X and Y variations at the same rate and an even number of stages in the multiplier will be avoided through proper scaling or use of an odd number of stages in the multiplier.

A.3 DIGITAL-OPERATIONAL QUASI-WORST-CASE ERROR FORMULAE FOR OPEN-LOOP INTEGRATION

A.3.1 Method of Analysis. A comprehensive error analysis of the DO Integrator in the general case has been initiated such that the developed error formulae would be usable to predict the errors for any DO mapping configuration. The generalized point of view excludes the use of an entirely deterministic pulse-by-pulse analysis. The analysis would become extremely involved as soon as the complexity of the program increased beyond 3 or 4 integrators. Consequently, it was decided to develop worst-case error term calculations, wherever applicable.

In the case of biased source errors, such as the truncation error, the worst case is defined as

(1) The integrand function Y(t) has the worst possible shape,

(2) The errors have the same sign during all the computation time; therefore, they must be added linearly at each iteration, and that the sign is not known; $a = \pm \Sigma a_k$.

In most cases, however, we have an "a priori" knowledge of the function Y(t); thus, the more Y(t) form factors are known, the more reduction can be taken on the worst-case error prediction. Statistical reduction is used for random errors and error propagation formulae. For instance, the roundoff errors are random by definition, being uncorrelated from one dY increment to the adjacent ones. Thus, only the standard deviation of this error term need be computed. How many standard deviation values approximate a "quasi-worst-case" error is not known at the present time; however, one standard deviation will be assumed as a first approximation of the error. The more that is known about the function to be implemented by the D.O. Program, the more reduction can be taken into the error prediction formulae.

A.3.2 Definition of Error Source Terms. These error terms correspond to the systematic errors introduced by the mechanization of integration in a D.O. integrator. They are independent of the actual problem mapping interconnections.

Two kinds of error terms will be considered on each connection of the map. First, the biased errors which can possibly keep the same sign during the total computation time will be estimated for the worst-case, linearly. Second, the random errors, uncorrelated from one interval to the next, will be estimated by their standard deviation.

In the formulae, the errors will always be expressed in number of pulse values, as a function of input errors, scale factors, estimated dynamic ranges, and form factors. The sign ± will always be assumed, unless specified.

A.3.2.1 Biased Error Source Terms

(a) Truncation Error ξ. This is the difference between the ideal z output and the actual output obtained when X(t) and Y(t) are the ideal input functions.

(b) Field Errors η_r and η_c. Errors introduced by the use of a smaller number of bits in the multiplier as compared to the number of bits of the X and Y counters.

(c) Integrand Biased Quantization Error ζ. This error is introduced by the use of a Y counter initial value (or constant) differing from the problem value by at most one l.s.b. (least significant bit).

A.3.2.2 Random Error Source Terms

Integrand Random Quantization Error ϵ. This error is due to the use of a quantized Y(t) function instead of a smooth ideal function.

A.3.3 Definition of Open-Loop-Error Propagation. In an open-loop program, the output pulses from any integrator do not feed back to the input of this integrator, even through several other operations. If this is the case, there

exists a simple relationship between the input errors to a given integrator, which are accumulated throughout the total computation time, and the output errors accumulated over the same time.

A.3.3.1 Open-Loop Biased Errors Propagation Terms

(a) Integrand input biased error propagated through the integrator, α_{yz}.

(b) Integration variable input biased error propagated through the integrator, α_{xz}.

A.3.3.2 Open-Loop Random Errors Propagation Terms

(a) Integrand input random error propagated through the integrator, β_{yz}.

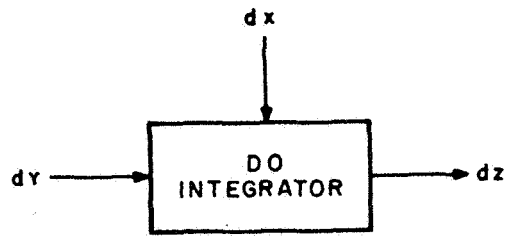
(b) Integration variable input random error propagated through the integrator, β_{xz}.

Figure A.3 summarizes the error source terms and the open-loop-error propagation terms.

A.3.4 Analytical Expressions

A.3.4.1 Scale Factors. A simplification in the designation of the scale factors is apparent when a basic scale factor of one (SC1) is assigned to an arbitrary power of two. This can be chosen, for convenience, as being the expected number of significant bits in the counters of the program. With this convention, a counter or connection marked with an SC1 designation, means that 2^n pulses will be required to operate a variation of one machine unit. Similarly, if the scale factor is $SC2^a$, it means that 2^{n+a} pulses will be required to operate the same variation of one machine unit.

The readout number of a counter, divided by the counter scale factor, will represent the real value of the counter, expressed in machine units. This definition has the advantage of using a unique setting of the scale factors, for a given program, independent of the possible tradeoff of accuracy vs. solution time, and the associated change in the register lengths. Moreover, the scale



dX INPUT	INPUT ERROR	α_x BIASED β_x RANDOM
dY INPUT	INPUT ERROR	α_Y BIASED β_Y RANDOM
dZ OUTPUT	OUTPUT ERROR	α_z BIASED β_z RANDOM

$$\alpha_z = \alpha_{xz} + \alpha_{yz} + \xi + \eta_R + \eta_C + \zeta$$

$$\beta_z = \beta_{xz} + \beta_{yz} + \epsilon$$

Figure A.3. DO Integrator Error Terms

factors remain small, since SC1 is the basic scale. This simplification is possible because it has been found that the term 2^n cancels out in all the input-output scale-factor relationships, and in most of the error formulae. It reflects the fact that the relative accuracy of a D.O. solution is directly proportional to the register length.

Figure A.4 illustrates the scale-factor equations.

A.3.4.2 Form Factors. In order to limit the worst-case error predictions, some reasonable assumptions have to be made on the shape of the functions Y(t) and eventually X(t), where t is the independent variable. Figure A.5 illustrates what would be a rough sketch of Y(t) and X(t), where a guess is made of the number and amplitudes of the monotonic variations. Three types of variations will be considered.

(a) Monotonic variations of X (j index),

(b) Monotonic variations of Y (i index),

(c) The interval without change of the sign of dX and dY (i j index).

The v_i's (variations of Y) and w_j's (variations of X) are expressed in machine units, so that the following values can be expressed as a first approximation.

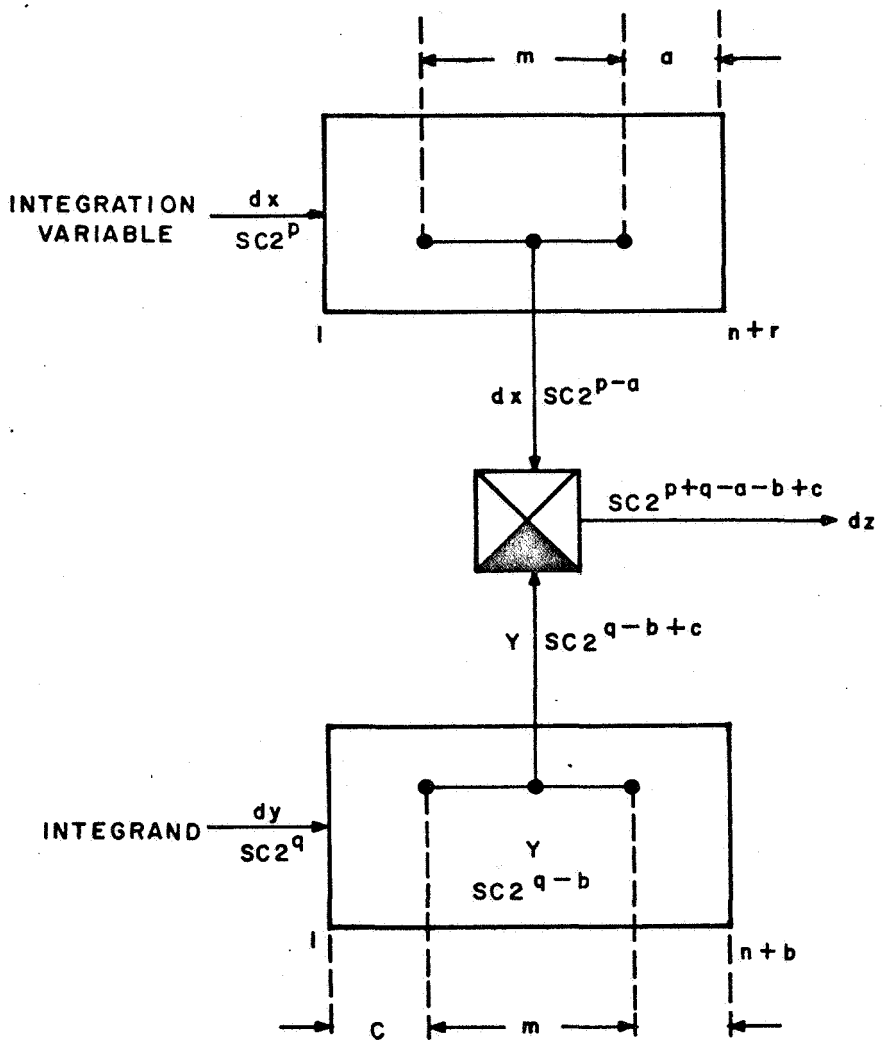
(a) One monotonic variation of Y = v_i expressed in machine unit on the dY input,

(b) One monotonic variation of X = w_j expressed in m.u. on the dX input,

(c) Total number of input pulses on dX $\mp 2^{n+p}\Sigma|w_j|$,

(d) Total number of input pulses on dY = $2^{n+q}\Sigma|v_i|$,

(e) Actual maximum value of the Y counter = Y_m. The Y counter when filled is equal to one and must always remain less than one to avoid overflow ,



NOTE :

THE X AND Y FIELDS (m STAGES) ADDRESSED TO THE MULTIPLIER HAVE EQUAL LENGTH INDEPENDENT OF THE COUNTER SCALE FACTORS.

Figure A.4. Scale Factors

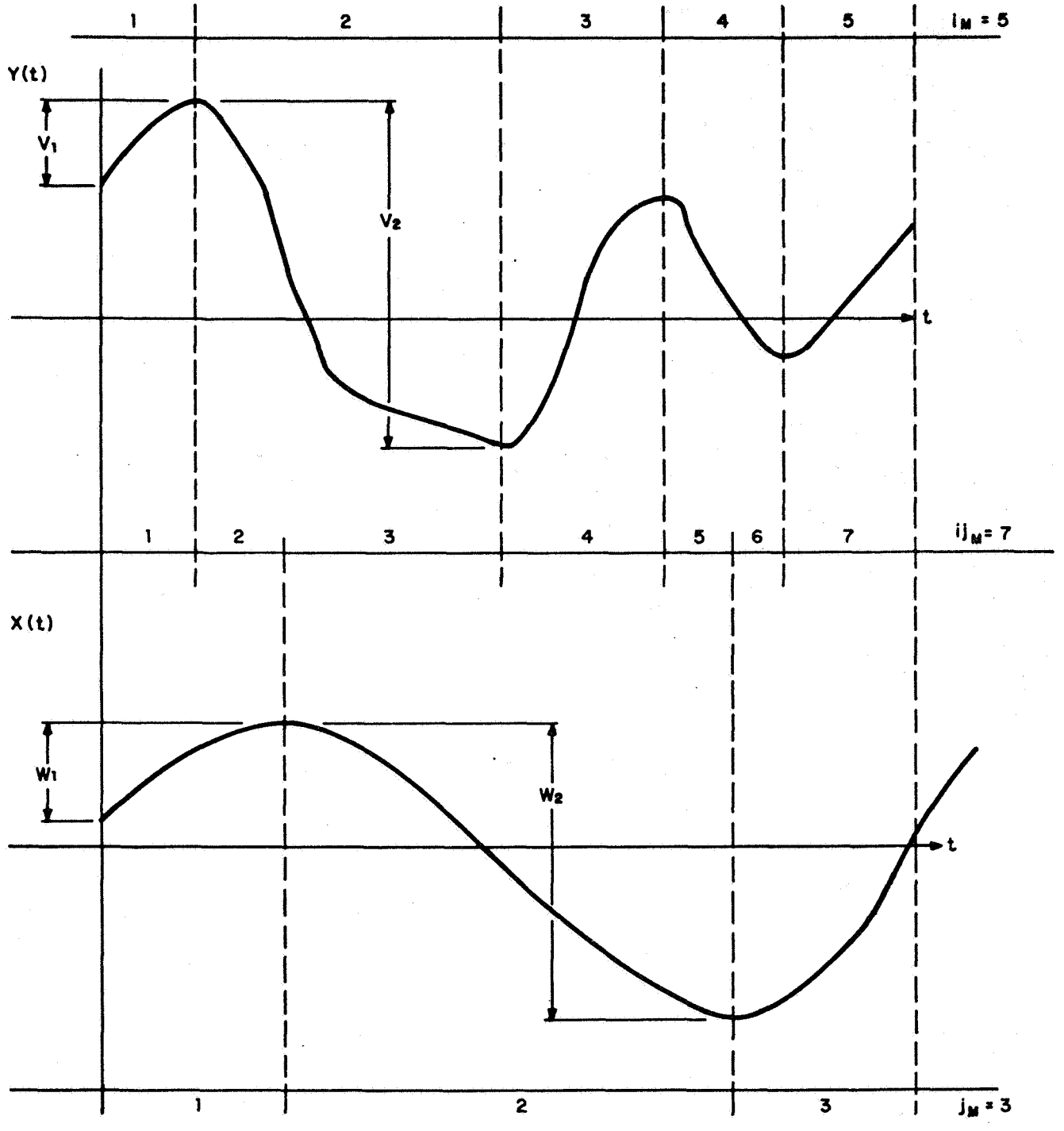


Figure A.5. Form Factors

(f) Average value of the Y counter

$$\bar{Y} = Y_o + \Sigma v_i (2^{q-b}) \quad ,$$

(g) Total variations of the Y counter

$$DY_t = 2^{n+q} \Sigma |v_i| \; 2^{-n-b}$$

$$= \Sigma |v_i| \; 2^{q-b} \quad .$$

A.3.4.3 Truncation Error. Refer to A.3.2.1.a for definition. It has been shown in A.2 that the maximum truncation error is equal to the number of bits of the multiplier, for a monotonic variation from zero to full-scale value of the Y counter, assuming that the worst condition of Section A.2.2 is avoided through use of an odd number of stages in the multiplier. However, the sign of this error term is unknown, as the same sign of X and Y variations may result in either a positive or negative error, depending upon the actual pulse-sequence order. According to the scaling conventions shown on Figure A.4,

$$\xi_{y \text{ monotonic}} = (m)\,(DY)\,(2^c) \quad , \tag{A-14}$$

will be used for monotonic variations of the integrand. However, if there are sign reversals of either X or Y increments, or if X goes beyond full-scale value, then, most likely, the sign of ξ will reverse. Finally, a statistical reduction of ξ will be used and rms values of ξ will replace worst-case linear accumulation.

$$\xi = \sqrt{\sum_{ij=1}^{k} [2^c \cdot DY_{ij} \cdot m]^2} \tag{A-15}$$

To this truncation error term should be added the initial value truncation error term, corresponding to the constant multiplier truncation error term. It has been shown in A.1 that this error term is null (0) for full-scale variations of the m field of the X counter, although it has a maximum value of

$$\xi_c = \pm\left(\frac{m}{6} + 1\right) \quad , \tag{A-16}$$

for a variation of the X counter larger than 1/4 of the m field of the X counter, that is, for $w_j > 2^{m-n+a-p-2}$.

A.3.4.4 Field Errors η_r and η_c. Refer to A.3.2.1.b for definition.

(a) X counter field error: η_r. An inspection of Figure A.4 shows that the higher transitions of the X counter, of an order higher than the highest order bit of the multiplier field, m, will not be used for the multiplier rate input. Consequently, the multiplier rate input scale factor was overestimated. For $2^{n+p} \Sigma w_j$ input pulses to the X counter, there will be $2^{n+p-a-m} \Sigma w_j$ higher-order transition not used by the multiplier. The total number of transitions normally used by the multiplier would be $2^{n+p-a} \Sigma w_j$. The field error, η_r, will be nonexistent when using a corrected scale factor at the multiplier rate input of

$$2^{p-a}(1 - 2^{-m}) \tag{A-17}$$

instead of 2^{p-a}. If the correction is not made, an output error η_r will be introduced

$$\eta_r = (1-2^{-m}) \Sigma v_i \Sigma w_j 2^{p+q-a-b+c} \tag{A-18}$$

It is assumed in the following that the correction will be made when necessary.

(b) Y counter field error: η_c. This error term is introduced by the quantization of the integrand in steps equal to the weight of the least significant bit of the m field in the Y counter. This term is identical in nature to the integrand random quantizqtion error ξ and will be evaluated in A.3.4.6.

A.3.4.5 Integrand Biased Quantization Error ζ. Refer to A.3.2.1.(c) for definition. The initial error in the Y counter of ± 1/2 pulse, will be accumulated at each increment of dX and, at the end of the computation time, the multiplier output error ζ will be

$$\zeta = \Sigma w_j \, 2^{n-m+p-a} \quad . \tag{A-19}$$

A.3.4.6 Integrand Random Quantization Error ϵ. Refer to A.3.2.2.(a) for definition. The quantization error for one step of the least-significant bit of an m bit multiplier has a variance

$$\sigma_{(\Delta Y)}^2 = \frac{2^{-2m}}{12} \quad , \tag{A-20}$$

where errors are supposed to be uniformly distributed within the quantization interval of ± 1/2 1sb of the multiplier field m. The quantization interval corresponds to $2^{n+b-m-c}$ input pulses to the Y counter. The multiplier output error during this interval, will be, on the average

$$\epsilon_{(\Delta Y)} = \frac{\sigma_{(\Delta Y)}}{\bar{S}}$$

where ϵ is the output error for a quantization interval ΔY and $\bar{S}$ is the average slope at the multiplier level:

$$\epsilon_{(\Delta Y)} = \frac{2^{-m}}{\sqrt{12}} \cdot \frac{2^{p-a} \Sigma |w_j|}{2^{q-b+c} \Sigma |v_i|}$$

$$\epsilon_{(\Delta Y)} = \frac{1}{\sqrt{12}} \frac{\Sigma |w_j|}{\Sigma |v_i|} \cdot 2^{p-q-m-a+b-c}$$

During the total computation time there are

$$\frac{2^{n+y} \Sigma |v_i|}{2^{n+b-m-c}} = 2^{m+q-b+c} \Sigma |v_i|$$

quantization intervals experiencing uncorrelated random errors. Therefore, the variance of the output error at the end of computation will be

$$\sigma_\epsilon^2 = \frac{1}{12} \left(\frac{\Sigma |w_j|}{\Sigma |v_i|} \right)^2 \left(2^{2(p-q-m-a+b-c)} \right) \left(2^{m+q-b+c} \Sigma |v_i| \right)$$

And a conservative estimate of the integrand random quantization error, ϵ, will be given by the standard deviation

$$\epsilon = \sigma_\epsilon = \frac{1}{\sqrt{12}} \frac{\Sigma |w_j|}{\sqrt{\Sigma |v_i|}} \left(2^{-\frac{m}{2} + p - \frac{q}{2} - a + \frac{b}{2} - \frac{c}{2}} \right) \qquad \text{(A-21)}$$

A.3.4.7 Open-Loop Propagated Biased Errors

(a) Y-counter input biased error propagated through the integrator: a_{yz}

The case of one monotonic variation of Y will be considered at first. The output error for the first error pulse of a_y will be

$$a_{yz_{(1)}} = 2^{n+p-a} \Sigma |w_j| \left(2^{-n-b} \right) \left(\frac{(a_y - 1)}{a_y} \right)$$

$$a_{yz_{(1)}} = \frac{a_y - 1}{a_y} \Sigma |w_j| \, 2^{p-a-b}$$

The output error for the k^{th} error pulse of a_y will be

$$a_{yz(k)} = \frac{a_y - k}{a_y} \Sigma |w_j| \, 2^{p-a-b}$$

The total output error at the end of computation will be

$$a_{yz} = 2^{p-a-b} \Sigma |w_j| \sum_{k=1}^{k=a_y} \left(1 - \frac{k}{a_y} \right)$$

with a_y integer and $a_y > 1$,

$$a_{yz} = 2^{p-a+b} \Sigma |w_j| \left(\frac{a_y - 1}{2}\right) \qquad \text{(A-22)}$$

This formula will now be extended to the case of driving functions with sign reversals. Similarly to that of truncation errors, in A.3.4.3, a statistical reduction of a_{yz} will be estimated by composition of the rms values of the errors evaluated for each monotonic interval i j.

$$a_{yz_{ij}} = 2^{p-a-b} |w_{ij}| \frac{a_{y_{ij}} - 1}{2}$$

and the total error will be

$$a_{yz} = \sqrt{\Sigma (a_{yz_{ij}})^2}$$

$$a_{yz} = 2^{p-a-b} \sqrt{\sum_{ij=1}^{k} w_{ij}^2 \frac{(a_{y_{ij}} - 1)^2}{4}} ,$$

$$a_{yz} = 2^{p-a-b-1} \sqrt{\sum_{ij=1}^{k} (w_{ij}^2 a_{y_{ij}}^2 + w_{ij}^2 - 2 w_{ij}^2 a_{y_{ij}})} , \qquad \text{(A-23)}$$

for a first approximation, $a_{y_{ij}} = a_y \frac{|w_{ij}|}{\Sigma |w_j|}$,

and Eq. (A-23) becomes

$$a_{yz} = 2^{p-a-b-1} \sqrt{\frac{a_y^2}{\Sigma w_j^2} \sum_{ij=1}^{k} |w_{ij}|^4 - \frac{2a_y}{\Sigma |w_j|} \sum_{ij=1}^{k} |w_{ij}|^3 + \sum_{ij=1}^{k} |w_{ij}|^2} . \qquad \text{(A-24)}$$

In most applications it suffices to consider the first term only

$$\alpha_{yz} = 2^{p-a-b-1} \frac{\alpha_y}{\Sigma |w_j|} \sqrt{\sum_{ij=1}^{k} |w_{ij}|^4} \quad . \tag{A-25}$$

(b) X-counter input biased error propagated through the integrator: α_{xz}

An average estimate of the output error is given by the formula

$$\alpha_{xz} = \bar{Y} \cdot \alpha_x \cdot 2^c \tag{A-26}$$

where the input error is assumed to be linearly distributed over the dX incrementation.

A.3.4.8 Open-Loop Propagated Random Errors.

(a) Y-counter input random error propagated through the integrator: β_{yz}

The input random errors can be considered as uncorrelated from one l.s.b. increment of the Y-register m field to the next increment. Considering that the DO multiplier circuit does not require temporary storage (the R-register found in DDA integrators), and that its output is an irregular function of its input; it can be assumed that a random error on its input will not be accumulated over the remaining computation times. Consequently, the output random error will be directly related to its input error through the average slope $\bar{S}$ at the multiplier level.

$$\beta_{yz} = \frac{\beta_y}{\bar{S}}$$

$$\beta_{yz} = 2^{p-q-a+b-c} \cdot \frac{\Sigma |w_j|}{\Sigma |v_i|} \cdot \beta_y \tag{A-27}$$

(b) X-counter input random error propagated through the integrator: β_{XZ}

An average estimate of the output error is identical to the biased error estimate

$$\beta_{XZ} = \overline{Y} \cdot \beta_X \cdot 2^C \qquad \text{(A-28)}$$

Formulae (A-26) and (A-28) are quite rough estimates but sufficient for most practical purposes. However, when the integration variable error contribution to the output error is known to be a function of final values only, $\overline{Y}$ should be replaced by Y final. This is the case for single function generation, such as sine-cosine, where the dX error variations take place within constant integrand values for which the functioning of the integrator is truly reversible.

A.4 D O CLOSED-LOOP PROPAGATED ERRORS

A.4.1 Definition. An iteration path links several integrators, constant multipliers, variable multipliers, and servos. A constant multiplier will be treated as an integrator and a variable multiplier as a pair of integrators. An open iteration path always begins with an "independent variable integrator"; that is; the primary input is the independent variable. If a subsequent module is also an independent variable integrator, then the iteration path is considered to pass through the integrator from the dy input to the dz output. If the module is a "dependent variable integrator", that is, one for which the primary input is a dependent variable, then the iteration path passes through the integrator from the dx input to the dz output. Passage through a servo is, of course, from the dy to the dz output. A closed iteration loop is obtained from an open iteration path by connecting the dz output of the last module to the dy input of the first module.

Most problems will involve several closed iteration loops. These loops will generally fuse at summing points and have common path segments.

A.4.2 Closed-Loop Gain Factor. A gain factor can be assigned to each loop for a given computation time T. This gain $G_{(T, L_k)}$ is defined to be the product of the following parameters: the average value $\bar{Y}_i$, during time T, of the integrand in each dependent variable integrator of the loop; the gains of the servos in the loop; and the gains of the independent variable integrators found along the loop.

$$G_{(T,\ L_k)} = \prod_{i=1}^{m}{}_{(T,\ L_k)}\ \bar{Y}_i \prod_{j=1}^{n}{}_{(T,\ L_k)}\ g_{T,\ S_d} \prod_{k=1}^{p}{}_{(T,\ L_k)}\ G_{T,\ I_k} \qquad \text{(A-29)}$$

The gains $g_{T,\ S_j}$ and $g_{T,\ I_k}$ are defined in the following subsections.

A.4.3 Servo Gain. The gain of a servo is the number of output pulses generated to balance one input pulse

$$g_{T,S_j} = \prod_{i=1}^{r} \frac{1}{\bar{Y}_{S_j,i}} \tag{A-30}$$

where the $\bar{Y}_{S_j,i}$ are the averages, occuring during the computation time T, of the values contained in the Y-counters of the integrators and constant multipliers found along the servo feedback path.

A.4.4 Independent Variable Integrator Gain. The gain of an independent variable integrator is the ratio of the number of output pulses to the number of Y-counter input pulses during the computation time T, assuming that the initial value in the Y-counter is equal to zero. This gain will depend upon the time of occurrence of the Y-counter input pulses. Two situations will be considered. In the first situation, all the input pulses occur within an initial time interval which is short with respect to the computation time T. The time of computation τ required for one input pulse to generate one output pulse is

$$\tau = \frac{1 \text{ mu ind. var (time)}}{2^{2(n-m)+p-a+b-c}} \tag{A-31}$$

If $a_{(T,INT)}$ and $b_{(T,INT)}$ are, respectively, the numbers of input and output pulses, then the gain of the integrator is

$$g_{T,I} = \frac{b_{(T,INT)}}{a_{(T,INT)}} = \frac{T}{\tau} \tag{A-32}$$

In the second situation, the input pulses are distributed throughout the time T. An approximation for the gain is

$$g'_{T,I} = \frac{b_{(T,INT)}}{a_{(T,INT)}} = \frac{T}{2\tau} \qquad \text{(A-33)}$$

If little is known about the pulse rate flow in the problem, formula (A-32) will be used for the highest derivative integrator, and formula (A-33) will be used for the lower-order derivative integrators.

A.4.5 Total Gain. A total gain can be defined at any Y-counter input, where two or more closed iteration loops merge, as the sum of the iteration loop gains.

$$G_{(T,\ Total,\ Loc.)} = \sum_{k=1}^{\ell} G_{(T,\ L_k)} \qquad \text{(A-34)}$$

It has been observed in several programs, where a large number of closed loops were involved, that the total gain is often negative. Basic closed-loop propagation error formulae (A-32) and (A-33) are applicable for positive as well as for negative total gains, when using appropriate intepretation and method of application.

A.4.6 Error Propagation in The General Case

A.4.6.1 Sources of Error. The error formulae developed in Section A-3 provide over-estimates of the errors generated in an open-loop integration scheme or in a closed-loop of iteration, when the time of computation T is too short for error pulses to propagate completely around the loop. In the more general case, the program contains several interacting closed iteration paths, and the time of computation is long enough for generation of significant truncation and biased quantization errors that start the error feedback process. An error value smaller than one pulse represents a shift of the time at which a variable is incremented but does not introduce an error in the number of pulses representing the quantized variable. The error propagation will be analyzed as a pulse flow superimposed on the normal signal pulse flow.

In a first attempt to determine the errors at the end of computation, in a program containing closed iteration paths, the following method was used. The error curve was considered as a slowly varying function of time; e.g., non-oscillating (this assumption was later found to be erroneous). An estimate of the error at the end of computation was then made at a given point of the loop, usually at the input to the highest derivative integrators. Then the errors were propagated throughout the map, using the α and the β formulae. Finally, a match was sought between the error initially assumed at the starting point of the loop with the error returned to this point by propagation. Successive corrections of the initial error assumption were applied in an attempt to achieve the match. The method has serious drawbacks. In the first place, the initial assumption is quite arbitrary, unless the order of magnitude of the error is known through some independent solution. In the second place, a match cannot usually be found since the propagated errors are overestimated. Consequently, a reduction factor of the propagated error should be used, taking into account over-estimates and feedback in the closed loop. A workable rule to take these factors into account has not been found despite numerous painstaking trials, and such a method of error matching has been abandoned. The following more realistic method, based on a closer inspection of the actual pulse flow in the program, was finally adopted.

In the first step an error propagation map is set up. This map makes it easy to recognize the closed iteration paths and includes indications of the $\bar{Y}_i$'s and g's, for the time of computation T. Such an error propagation map may or may not be decomposed into isolated closed loops, depending upon the programmer needs for visual aids. Let us assume the loops have been isolated and numbered, loop 1, 2 ...

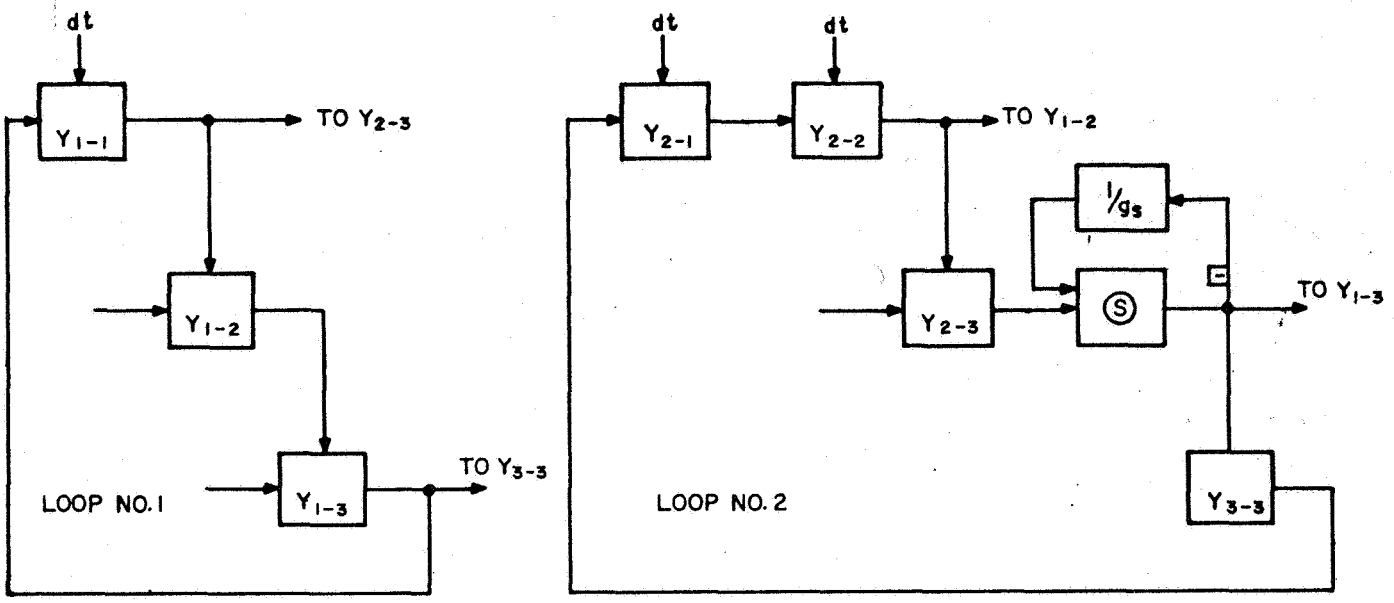


Figure A.6. Coupled Closed Iteration Loops

and let us consider the example illustrated in the first two interconnected loops of Figure A.6. In each loop, the errors will be first generated, then propagated, then fed back to the point of generation. Two kinds of error propagation will take place. In primary error propagation, the error is propagated within a single loop independently of other loops. In secondary error propagation, the errors transfer from one loop to the dy inputs of another loop, as shown in Figure A.6. A rigorous quantitative determination of the error growth process will require the following procedure.

A.4.7 Procedure For Closed-Loop Error Evaluation

A.4.7.1 Establish the error propagation map and isolate the various closed loops of iteration.

A.4.7.2 Slice the total computation time in intervals T_1, $T2$ ... T_n in such a way that within each interval no integrand or gain varies significantly (not more than 50 % change in value from one interval to the next). Indicate the $\overline{Y}$'s and g's on the error propagation map for the successive computation time intervals.

A.4.7.3 Compute the source errors for the interval of computation T_1 for all points of the map, using open loop error formulae of Section A.1. However, very low gain loops may be omitted since they will not contribute significantly to the final error. In many instances field errors and quantization errors may also be neglected when compared to biased quantization errors and truncation errors.

A.4.7.4 Then evaluate the primary error propagation during computation interval T_1 for each loop, using the formulae A-26, A-28 and A-31, A-32 or A-33. Each error term should be propagated only once around the loop. If at any input to an independent variable integration, the sum of the propagated errors is at least one pulse, then determine both the amplitude, a, or minimum number of pulses that this input error must attain and the time t_r required in order for this error to be compensated by the same number of error pulses fed back with opposite sign (assuming negative loop gain). The amplitude, a, will be one pulse unless the use of upscaling servos or other devices implies that the minimum must be more than one pulse. The time t_r will be called the error relaxation time since it corresponds to an oscillation of an error of amplitude a at this point of the loop and of repetition rate t_r. However, when other causes of error are taken into consideration, this oscillation may not be observable, although error growth is still limited to the maximum that can be reached during the time t_r. In particular, t_r varies from one loop to another and will change at the next computation interval, T_2.

In the case of positive feedback, the same time, t_r, will be computed, since it corresponds to a minimum time of error incrementation through feedback. New error increments will be computed for each t_r time span until the end of the computation interval T_1 in order to evaluate the error build up.

A.4.7.5 Evaluate, at the end of T_1, the errors introduced in a given loop from the other loops through the dy inputs of the given loop, taking into account the primary propagated error reduction due to error relaxation. When there

is relaxation, the sign of these errors is unknown, and the sign should be chosen for the worst case.

A.4.7.6 To these secondary propagated errors, add the ordinary sources of error generated during T_2 such as delays, truncation, etc. and compute the primary propagated errors in a way similar to that just described for T_1 in A.4.7.4. Then evaluate the secondary propagated errors at the end of T_2 in order to start the evaluation during T_3 and so forth until T_n in reached. A conservative estimate of the errors will then require summation of the error values found at each computation interval T_1, $T_2 \ldots T_n$. Very often, error relaxation will be found and the sign of the errors will not be known.

However, for medium and large size problems, this error-prediction method will not be amenable to hand calculations because of the large number of operations involved. Each error source term has to be propagated around a loop for each computation interval. During the first interval, T_1, only a few significant error sources may be found; however, for the next computation interval, secondary propagated errors are considered and as many error sources will be found as there are integrators in the problem. A general-purpose computer program could be set up to solve this error prediction problem, which would require a special coding of the map and numerous but simple algebraic equation-solving routines. However, it would be an important and complex task well beyond the scope of this study.

Fortunately, it has been found that in cases where the errors grow to a level of several pulses during the computation time, the secondary propagated errors can be neglected. The condition of error relaxation for negative feedback, or error buildup for positive feedback, will be found in the highest-gain closed loops which will be treated as error driving loops. A simplified procedure amenable to simple hand calculation can be set on this basis. However, these methods are not easily applicable to the free-running sine-cosine generator where the error sources are significantly larger than the propagated error buildup and a specialized error propagation analytical model was derived for this case, as discussed in Section 2.0.

APPENDIX B

ERROR CURVES

MEDOC PROGRAM RUNS

This section is concerned with the results of Task II, MEDOC Simulation. Tables B.1 through B.8 illustrate the maximum pulse errors for every program run. Also, included are error curves that are pertinent to the discussions in Sections 2, 3 and 5.

Those programs in which error curves are included in this report are identified in the tables by an index mark Δ.

The Reader can easily calculate the percent error of full scale:

$$\% \ \epsilon = \text{Pulse Error} \ (3.05 \times 10^{-3}) \quad . \tag{B-1}$$

B.1 PROGRAM SET 1.000 – ONE-AXIS INVESTIGATION

B.1.1 Initial Conditions

Program	a_1	a_3	Euler Angles Radians
1.100	1.0000	0.0000	$\gamma = \psi = \phi = 0$
1.200	0.8314 9663	0.5555 2980	$\gamma = 0.589$
			$\psi = \phi = 0$

B.1.2 Program Runs

A. Rotation

	Direction
1.101	CW(+)

B. Limit Cycle

	Radian Amplitude (±)
1.103	0.0000305
1.104	0.000244
1.105	0.001953
1.106	0.015625
1.107, 1.202	0.125
1.108, 1.203	1.000
1.109, 1.201	2.000

B.1.3 Table of Results for Maximum Pulse Errors

A. Rotation

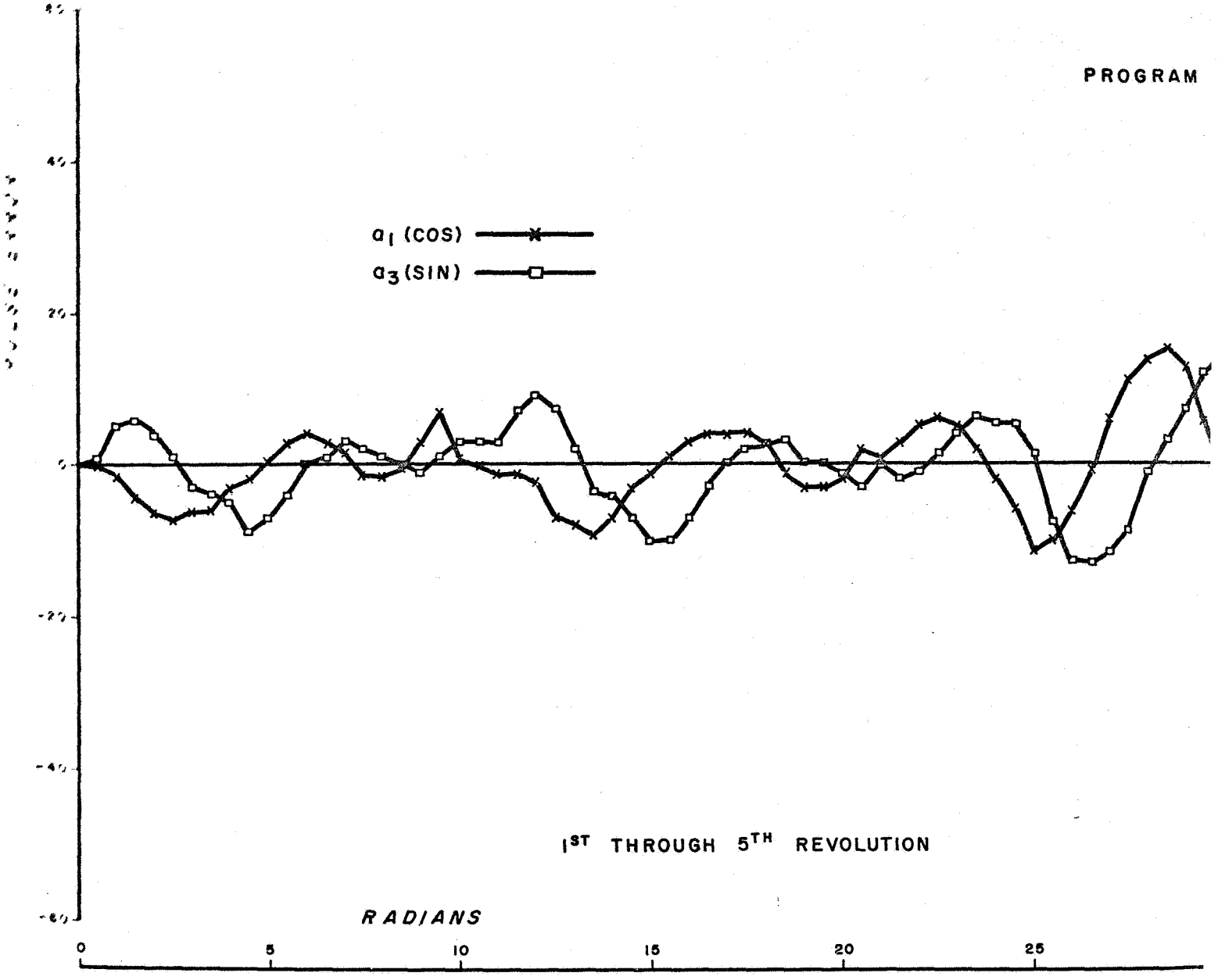
Table B.1

Program Run	Maximum Pulse Errors a_1	a_2	a_3	Rev.
	7		9	1
1.101	7		9	2
Δ	9		10	3
	11		6	4
	16		16	5
	19		19	10
	42		47	20

B. Limit Cycle

Table B.2

Program Run	Maximum Pulse Errors			Program Run	Maximum Pulse Errors			Cycle
	a_1	a_2	a_3		a_1	a_2	a_3	
	0		1		0		1	1
1.103	0		1	1.104	0		1	2
	0		1		0		1	10
	0		1		0		1	20
	0		1		1		0	1
1.105	0		1	1.106	1		0	2
	0		1		1		0	10
	0		1		1		0	20
	1		1		1		1	1
1.107	1		1	1.202	1		1	2
	1		1		1		1	10
	1		1		1		1	20
	4		5		7		5	1
1.108	4		6	1.203	6		7	2
	7		8		11		10	10
	5		7		23		22	20
	6		7		5		5	1
1.109	7		7	1.201	4		7	2
Δ	13		13	Δ	13		15	10
	19		20		21		25	20


PROGRAM
a_1 (COS)
a_3 (SIN)
1ST THROUGH 5TH REVOLUTION
RADIANS
0
5
10
15
20
25

GRAM RUN 1.101

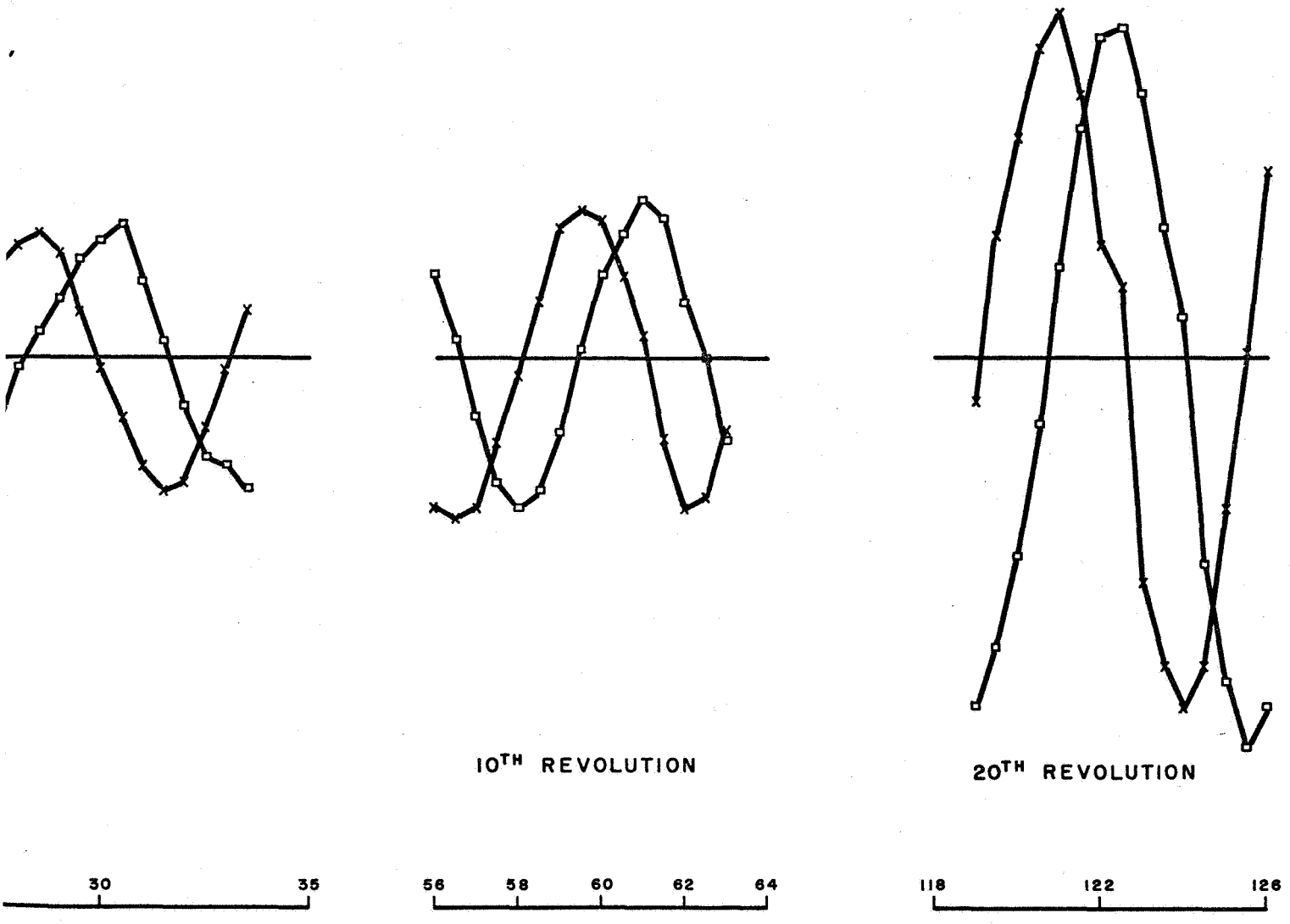
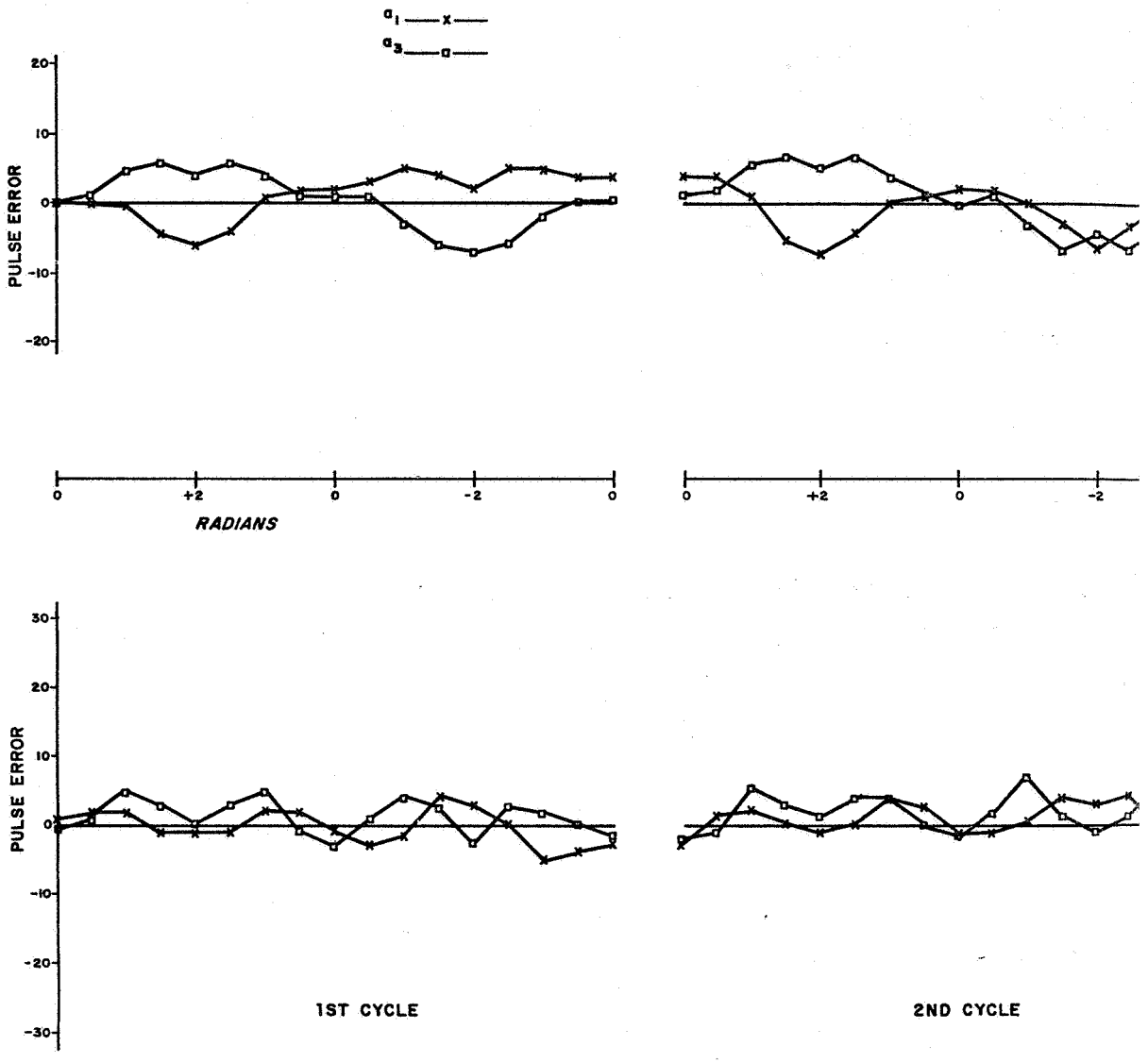


Figure B.1


a_1 —x—
a_3 —□—
PULSE ERROR
20
10
0
-10
-20
0
+2
0
-2
0
RADIANS
PULSE ERROR
30
20
10
0
-10
-20
-30
1ST CYCLE
2ND CYCLE

a

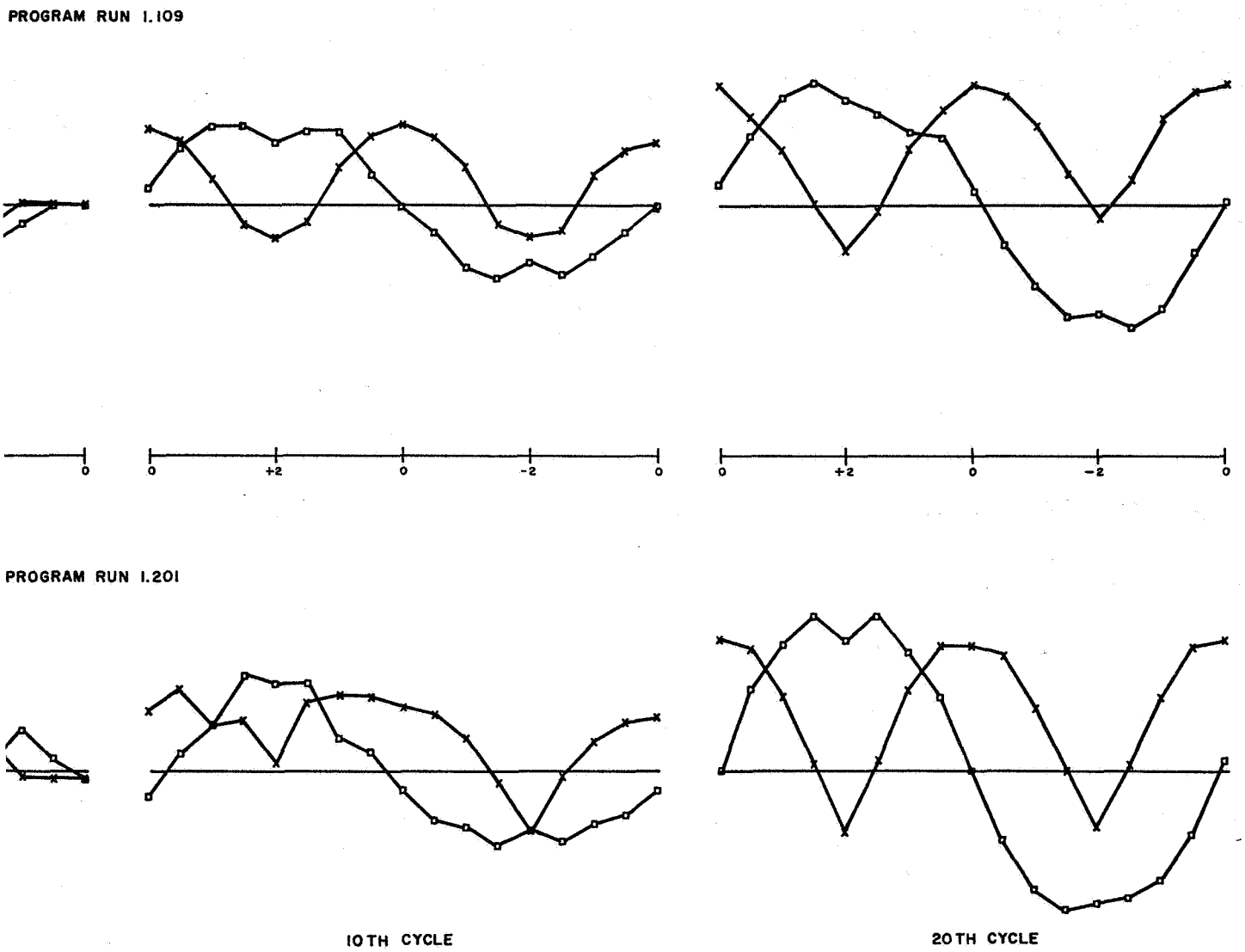

PROGRAM RUN 1.109
0
0
+2
0
-2
0
0
+2
0
-2
0
PROGRAM RUN 1.201
10TH CYCLE
20TH CYCLE

Figure B.2

B.2 PROGRAM SET 2.000 – TWO-AXIS INVESTIGATION

B.2.1 Initial Conditions

Program	a_1	a_2	a_3	Euler Angles (Radians)
2.100	.7316 8887	.4646 3139	.4987 4749	$\gamma = \psi = 0.75$ $\phi = 0$

B.2.2 Program Runs

A. Rotation

	Direction	Relative Angular Rates
2.101	CW(+)	$\Delta\theta_2 = \Delta\theta_1$
2.102	CCW(-)	$\Delta\theta_3 = 0$
2.103	CW(+)	$\Delta\theta_1 = 1/2\Delta\theta_2$ $\Delta\theta_3 = 0$

B. Limit Cycle

	Radian Amplitude (±)	Relative Angular Rates
2.104	1	$\Delta\theta_2 = \Delta\theta_1$
2.105	1/8	
2.106	1/2	$\Delta\theta_3 = 0$
2.107	1/8	$\Delta\theta_1 = 1/2\Delta\theta_2$
2.108	1/2	
2.109	1	$\Delta\theta_3 = 0$

B.2.3 Table of Results for Maximum Pulse Errors

A. Rotation

Table B.3

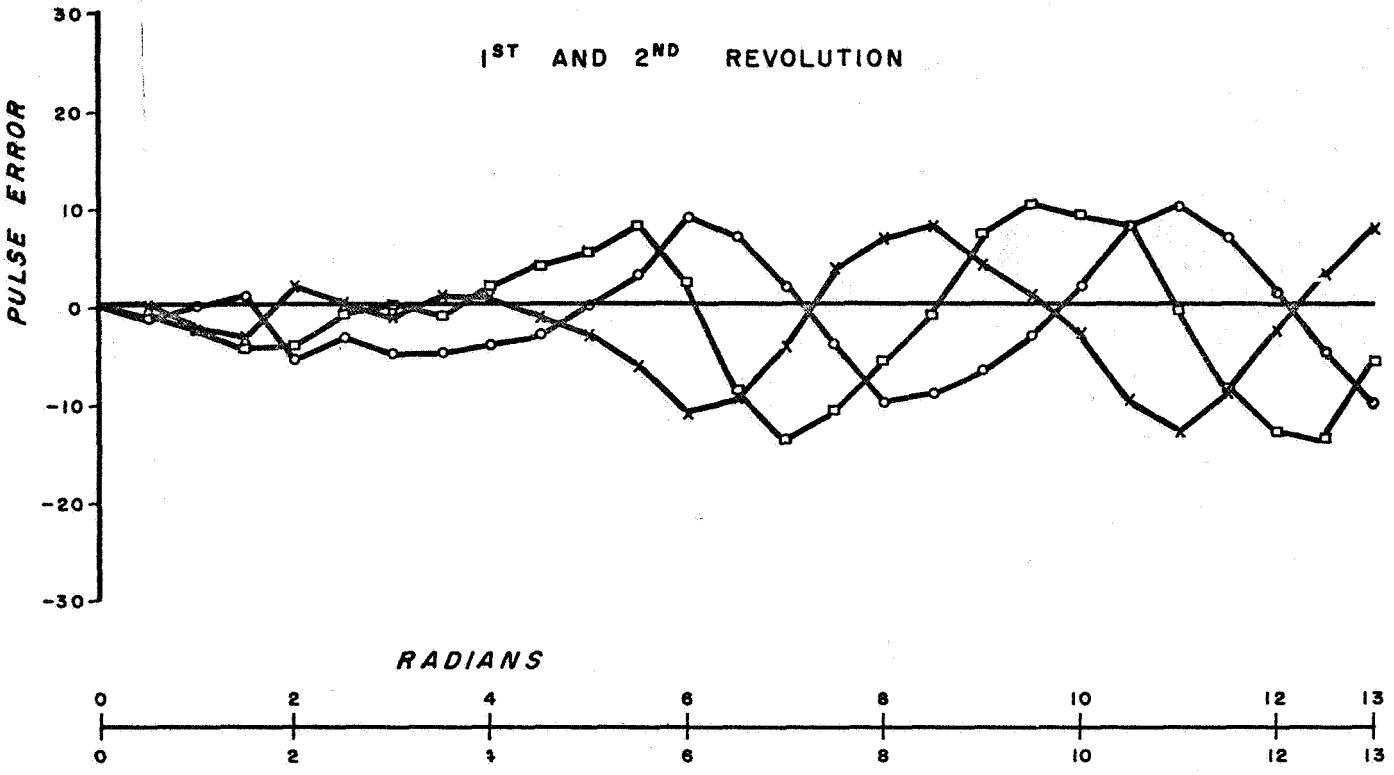
Program Run	Maximum Pulse Errors			Program Run	Maximum Pulse Errors			Rev.
	a_1	a_2	a_3		a_1	a_2	a_3	
	11	9	8		4	6	6	1
2.101	13	10	14	2.103	12	6	9	2
	18	19	18		55	42	52	10
Δ	18	19	27		63	60	54	20
	4	5	6					1
2.102	7	7	8					2
Δ	7	12	8					10
	29	31	43					20

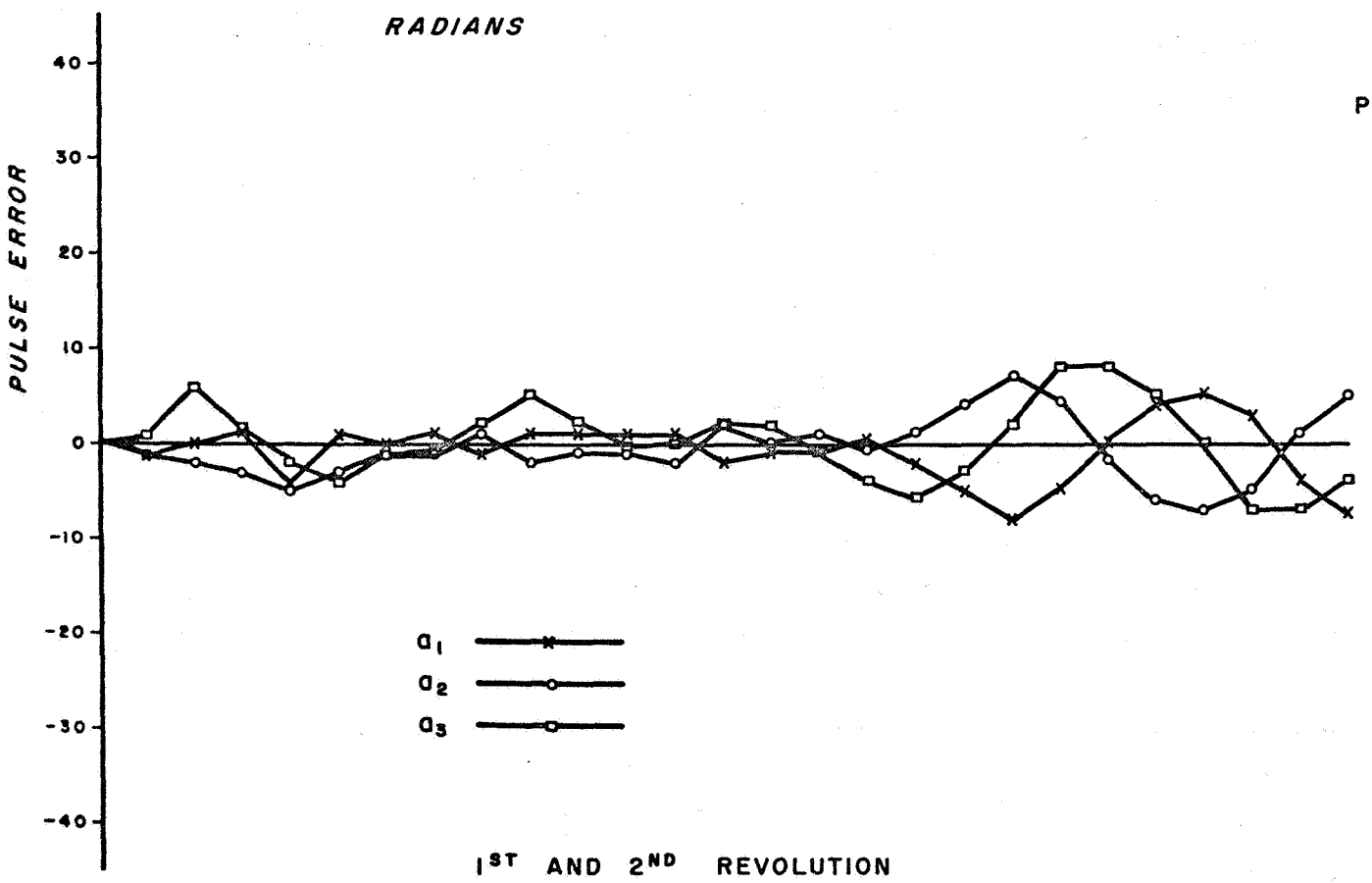
B. Limit Cycle

Table B.4

Program Run	Maximum Pulse Errors			Program Run	Maximum Pulse Errors			Cycles
	a_1	a_2	a_3		a_1	a_2	a_3	
	2	3	3		3	2	2	1
2.104	2	3	6	2.109	3	2	2	2
	7	8	8		11	5	9	10
	20	25	35		14	6	10	20
	1	1	2		1	2	2	1
2.105	1	1	3	2.107	2	2	1	2
	1	2	6		2	3	2	10
	1	5	9		7	10	3	20
	2	2	3		2	3	2	1
2.106	2	1	3	2.108	3	3	4	2
	7	7	12		6	3	4	10
	10	9	16		8	6	4	20

PROGRAM



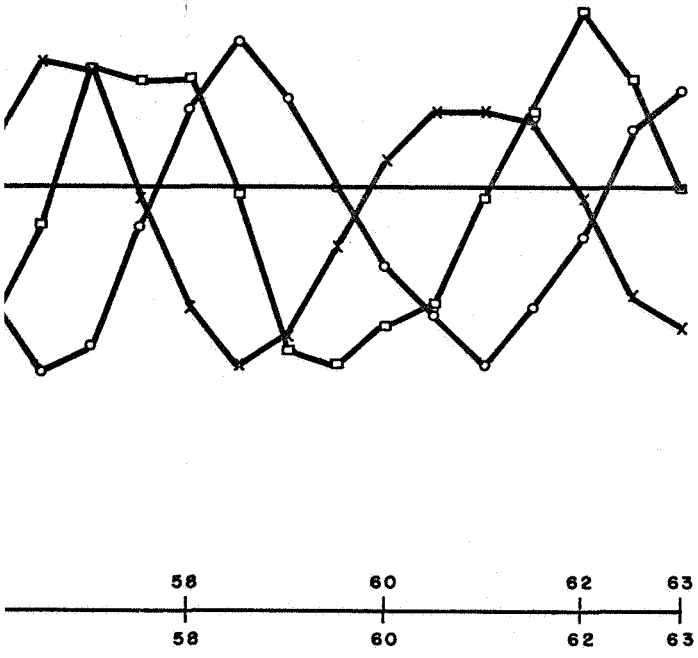


PROGRAM

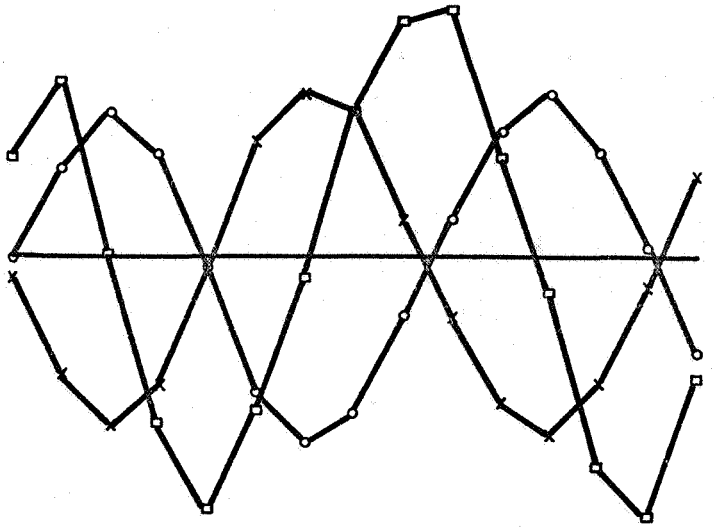
B

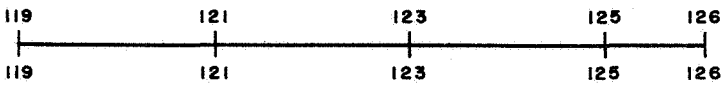
RUN 2.101

10TH REVOLUTION

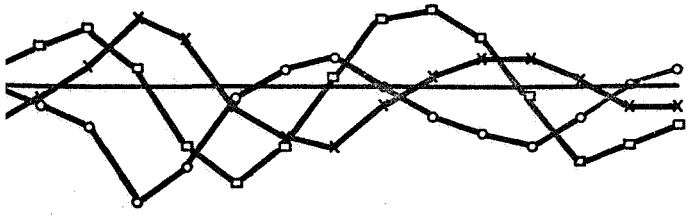


20TH REVOLUTION

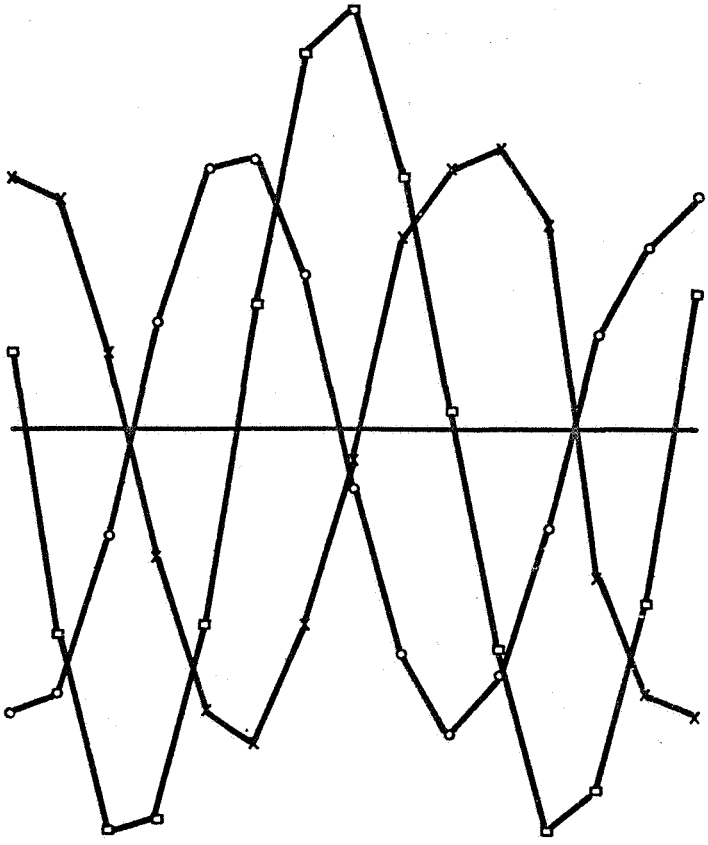




RUN 2.102



10TH REVOLUTION



20TH REVOLUTION

Figure B.3

B.3 PROGRAM SET 3.000 – THREE-AXIS INVESTIGATION

B.3.1 Initial Conditions

Program	a_1	a_2	a_3	Euler Angles (Radians)
3.100	+.5353 6860	-.1587 8187	+.8295 5939	$\gamma = \psi = \phi = .75$
3.200	+.7701 5115	-.2190 2415	+.5990 7898	$\gamma = \psi = \phi = .50$

B.3.2 Program Runs

A. Rotation

	Direction	Relative Angular Rates
3.101, 3.201	CW(+)	$\Delta\theta_3 = \Delta\theta_2 = \Delta\theta_1$
3.102	CCW(-)	
3.108, 3.208	CW(+)	$\Delta\theta_2$ = Reference
3.109	CCW(-)	$\Delta\theta_1 = 1/2\Delta\theta_2$; $\Delta\theta_3 = 1/4\Delta\theta_2$

B. Limit Cycle

	Radian Amplitude (±)	Relative Angular Rates
3.103, 3.202	1	
3.104	1/8	
3.105	1/64	$\Delta\theta_3 = \Delta\theta_2 = \Delta\theta_1$
3.106	1/16	
3.107, 3.203	1/32	
3.110, 3.210	1	
3.111	1/8	$\Delta\theta_2$ = Reference
3.112	1/64	$\Delta\theta_1 = 1/2\Delta\theta_2$
3.113	1/16	$\Delta\theta_3 = 1/4\Delta\theta_2$
3.114, 3.211	1/32	

B.3.3 Table of Results for Maximum Pulse Errors

A. Rotation

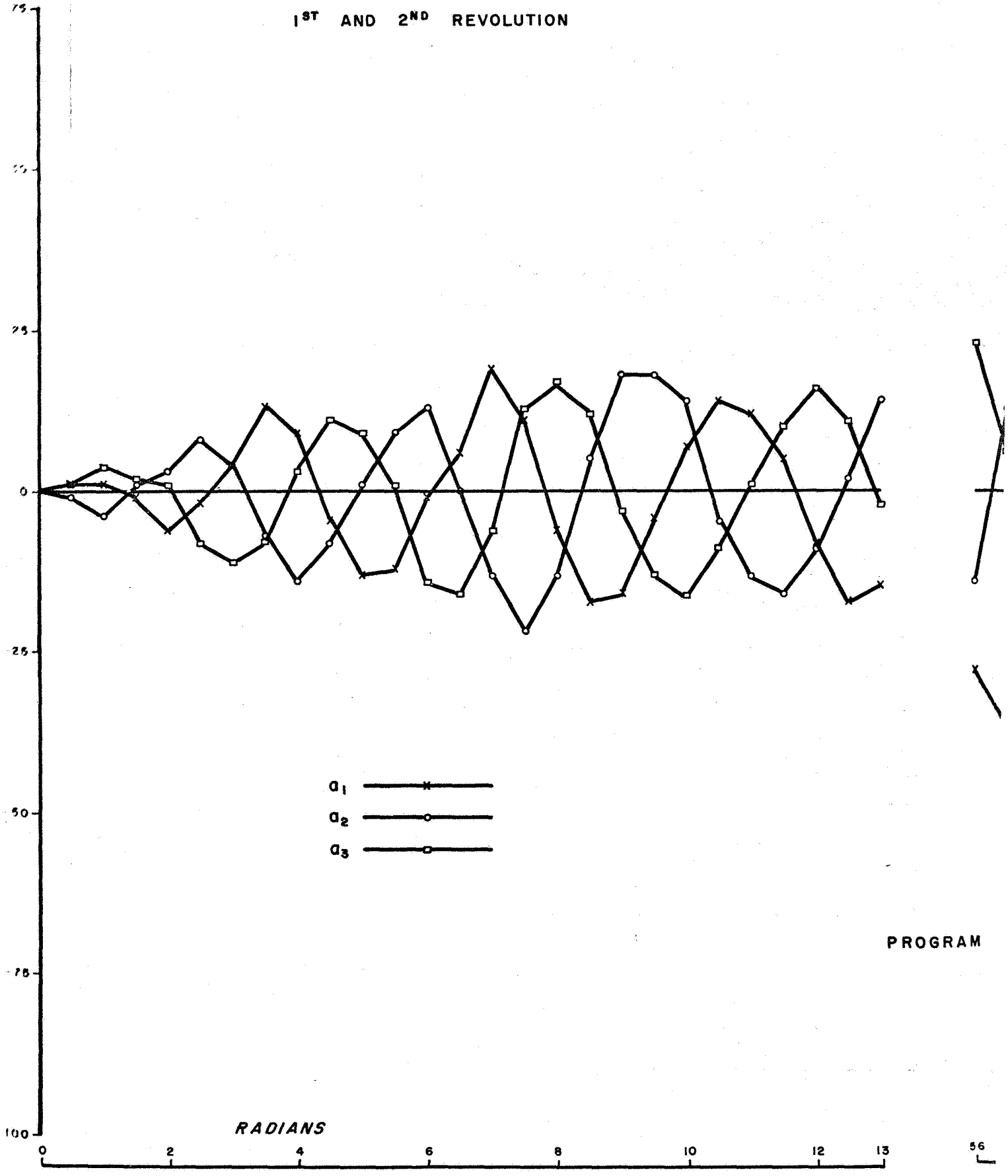
Table B.5

Program Run	Maximum Pulse Errors a_1	a_2	a_3	Program Run	Maximum Pulse Errors a_1	a_2	a_3	Rev.
	13	14	15		13	15	13	1
3.101	19	22	17	3.201	10	14	11	2
Δ	36	34	34		36	37	39	10
	93	91	92		50	49	52	20
	9	11	10		6	3	7	1
3.102	10	14	13	3.109	12	7	12	2
Δ	31	31	44		59	43	52	10
	40	37	38		105	94	100	20
	8	7	9		6	6	8	1
3.108	14	17	12	3.208	9	8	10	2
Δ	40	37	44	Δ	78	66	76	10
	61	53	61		186	87	211	20

B. Limit Cycle

Table B.6

Program Run	Maximum Pulse Errors a_1	a_2	a_3	Program Run	Maximum Pulse Errors a_1	a_2	a_3	Cycles
	4	4	4		6	5	7	1
3.103	7	16	5	3.202	6	6	4	2
	20	16	13	Δ	21	15	14	10
Δ	23	27	16		8	10	9	20
	1	1	0		1	2	0	1
3.104	1	1	2	3.111	2	2	1	2
	2	6	5		1	1	3	10
Δ	2	10	3		1	2	6	20
	0	1	1		0	1	0	1
3.105	0	1	1	3.112	0	1	0	2
	0	1	1		2	1	1	10
	0	1	1		2	1	0	20
	1	1	1		0	1	3	1
3.106	2	1	1	3.113	1	1	2	2
	3	1	1		0	3	2	10
Δ	4	4	3		1	5	4	20
	1	1	1		1	1	1	1
3.107	1	1	1	3.203	1	1	1	2
	2	3	2		1	5	1	10
Δ	5	3	0	Δ	2	5	3	20
	4	3	5		2	3	3	1
3.110	2	3	6	3.210	5	6	6	2
	13	5	15		10	4	7	10
	20	13	23		10	4	12	20
	1	1	0		0	1	1	1
3.114	1	1	1	3.211	0	1	1	2
	1	2	5		0	1	1	10
	0	2	3		0	1	1	20


1ST AND 2ND REVOLUTION
0
25
50
75
100
a_1
a_2
a_3
RADIANS
0
2
4
6
8
10
12
13

PROGRAM

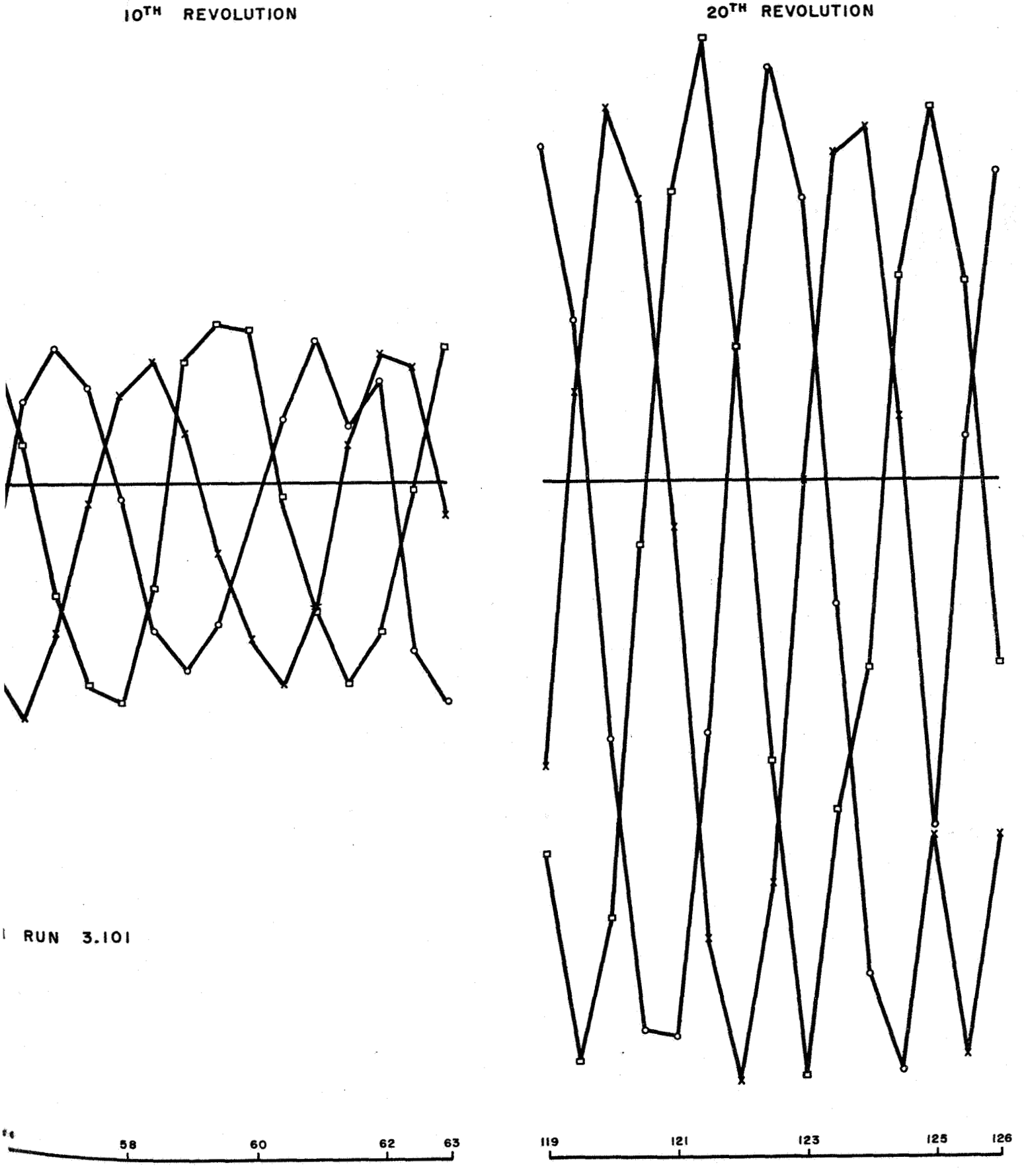
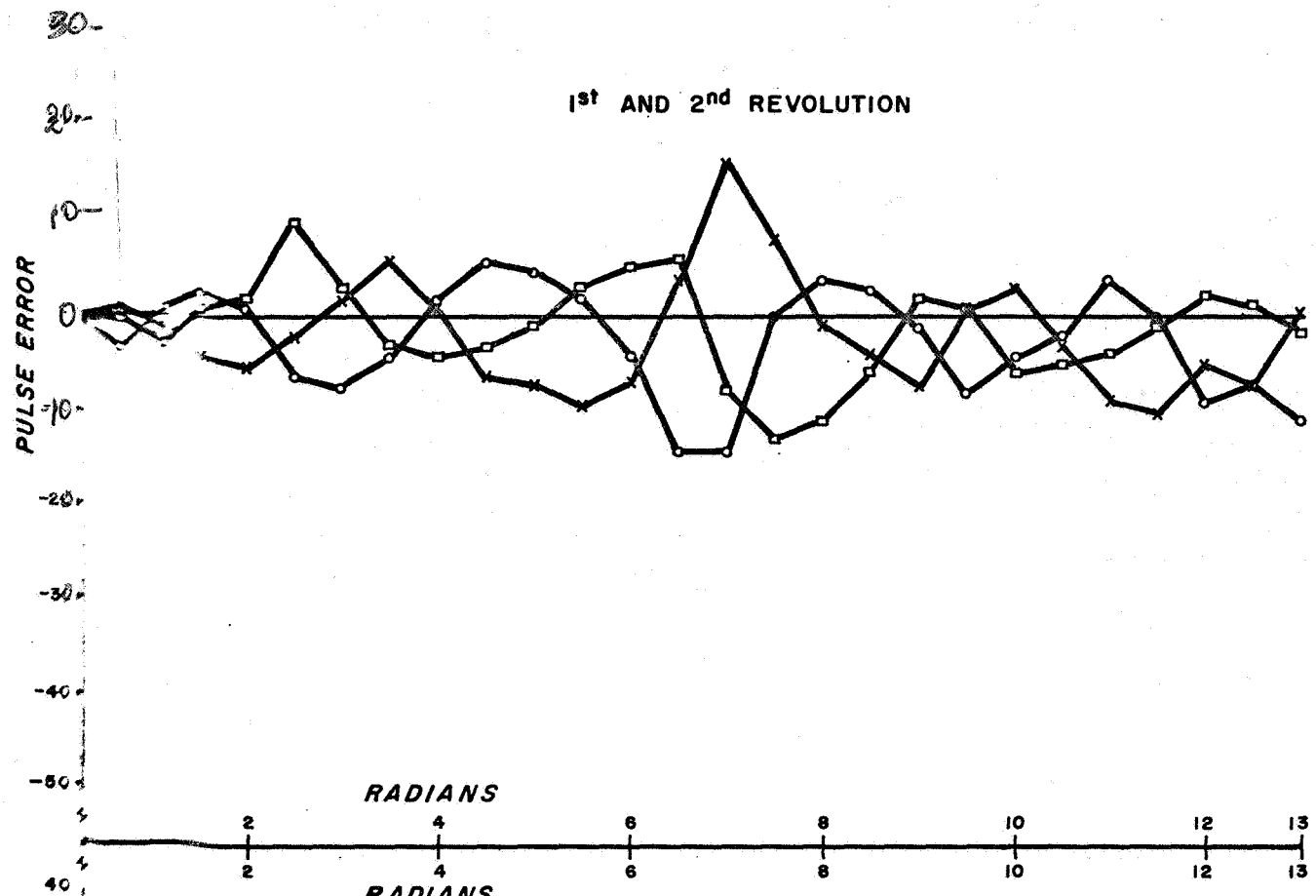
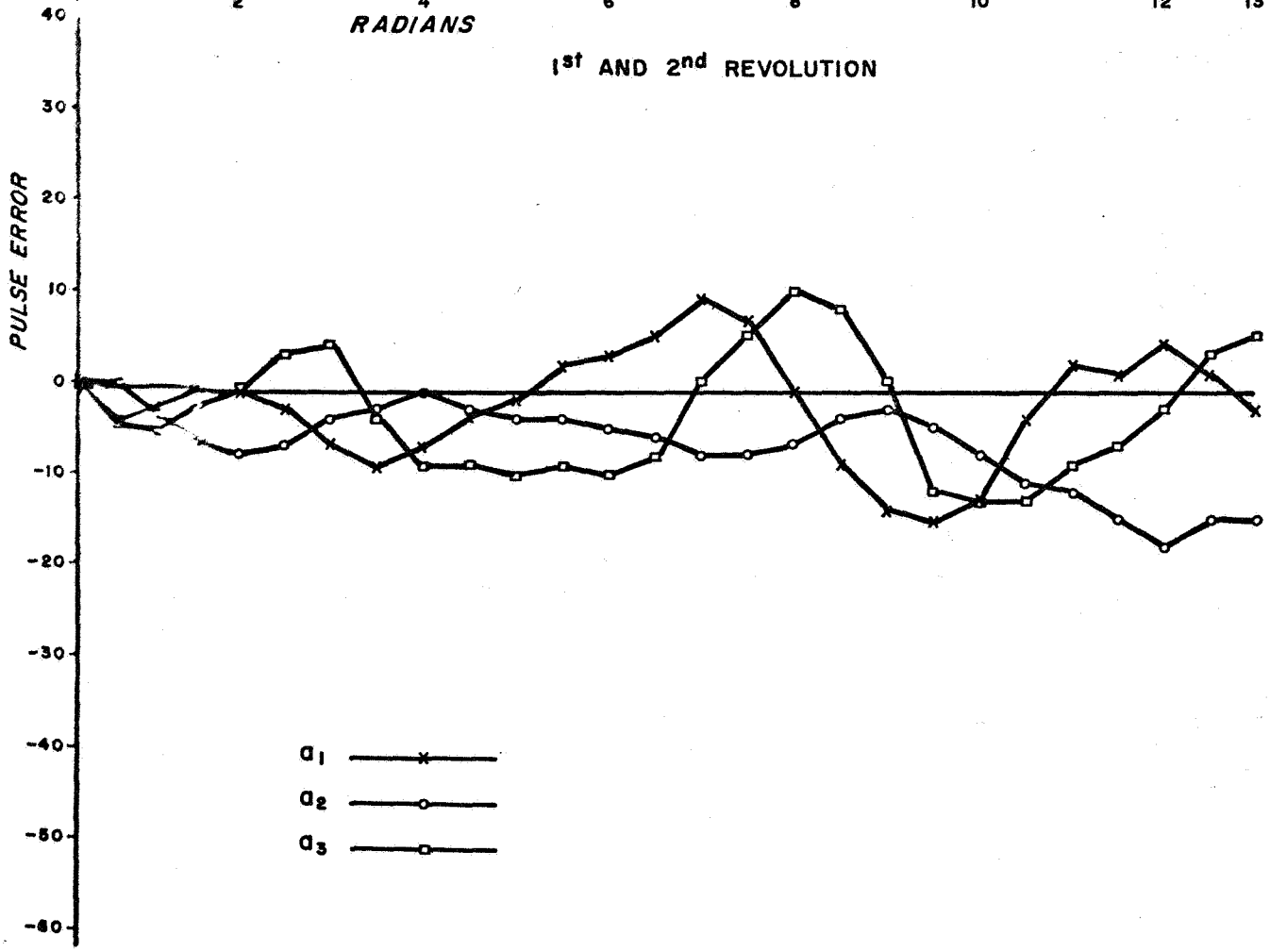

10TH REVOLUTION
20TH REVOLUTION
RUN 3.101
58
60
62
63
119
121
123
125
126

Figure B.4



PROGRAM RI

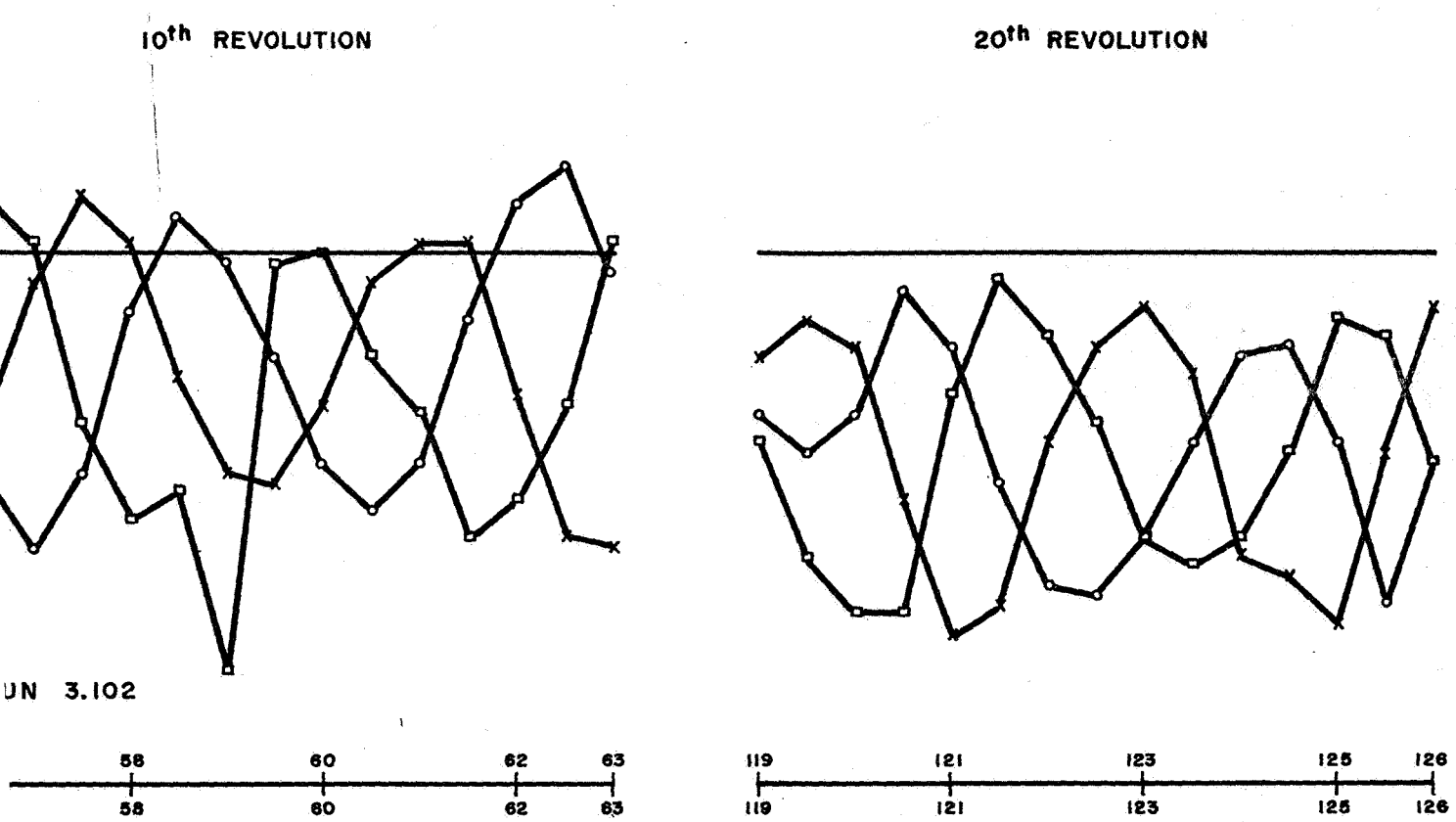
56



PROGRAM RU

A

e


10th REVOLUTION
20th REVOLUTION
UN 3.102
58
60
62
63
119
121
123
125
126

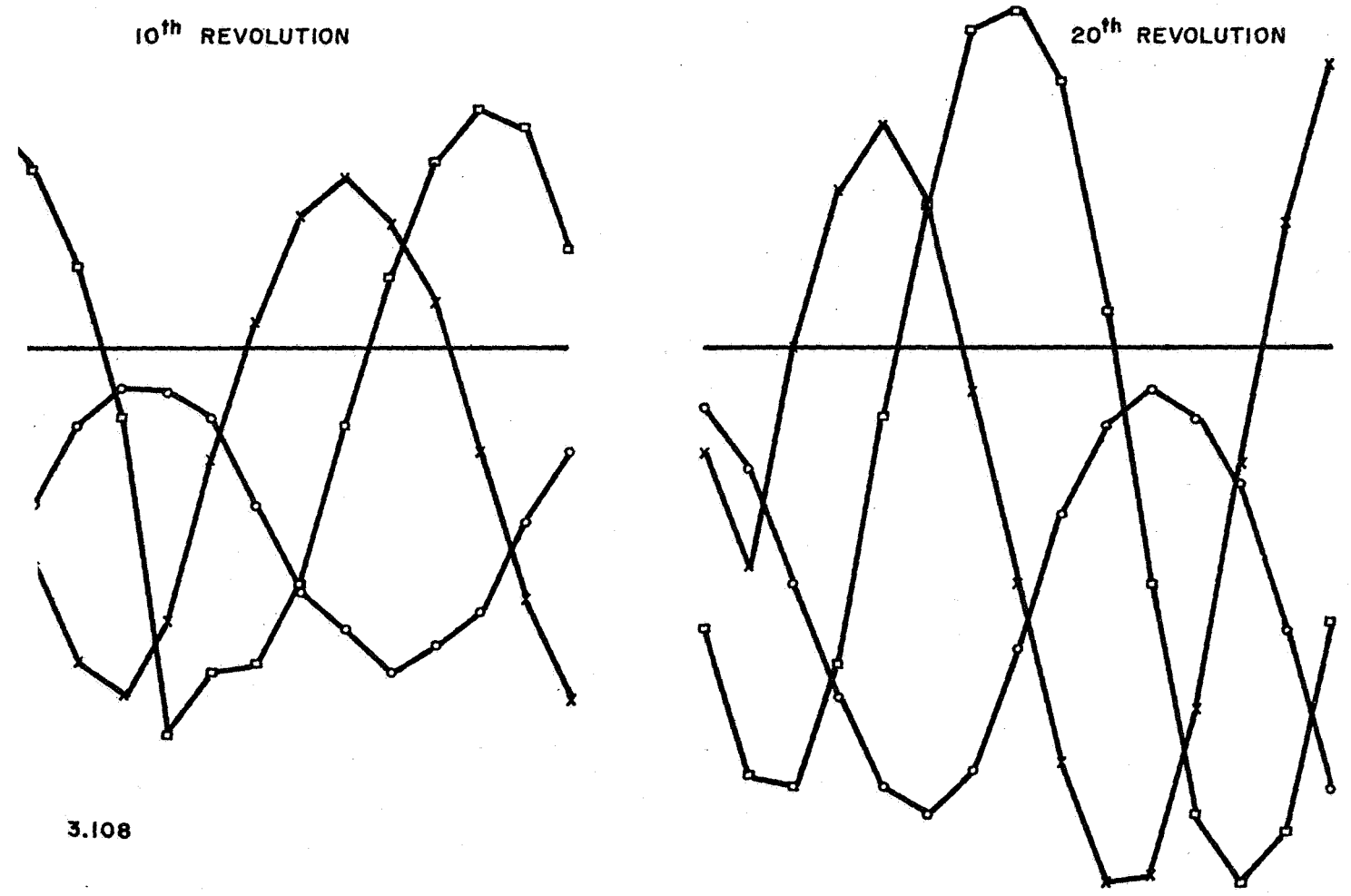
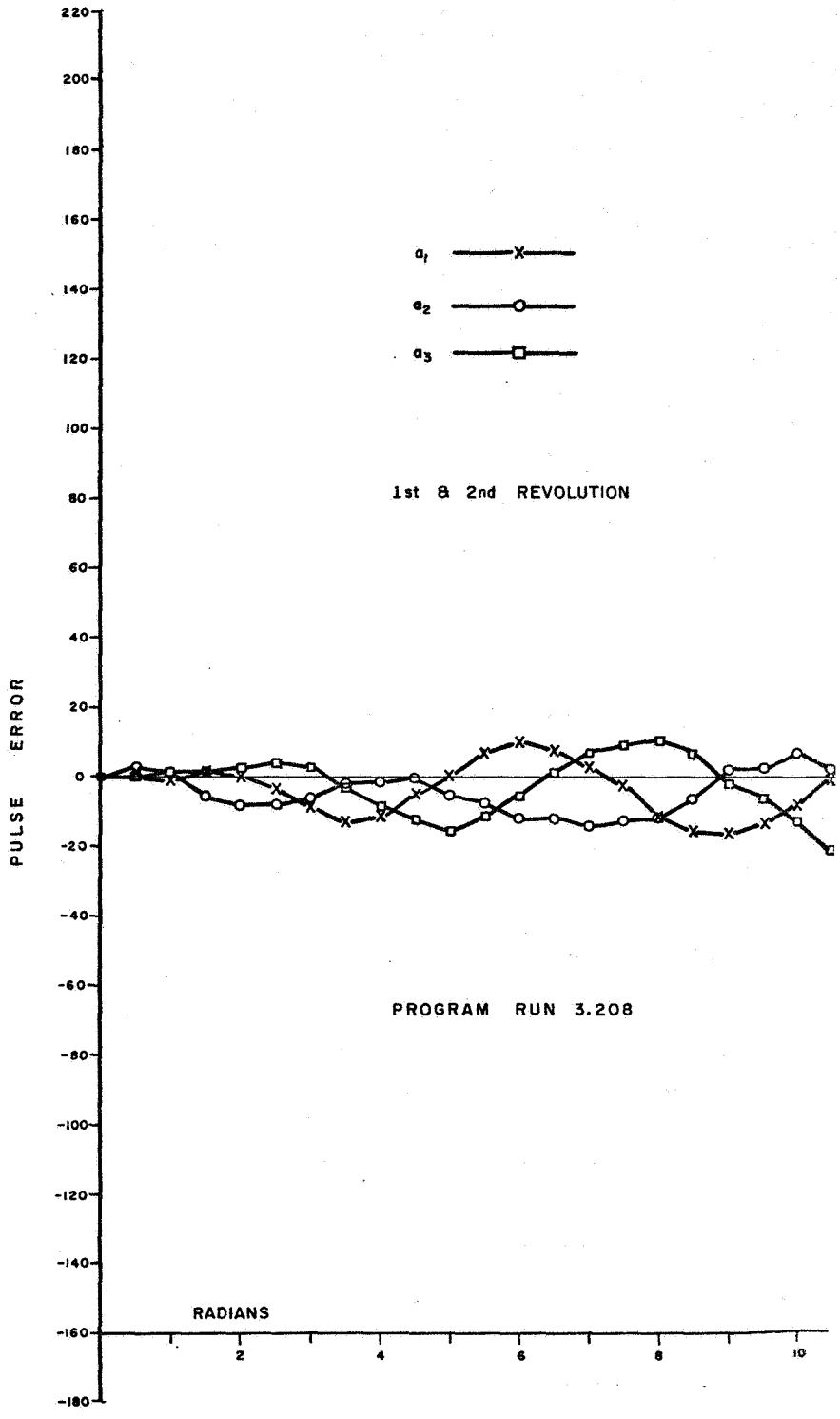

10th REVOLUTION
20th REVOLUTION
3.108

Figure B.5


220
200
180
160
140
120
100
80
60
40
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
PULSE ERROR
a_1
a_2
a_3
1st & 2nd REVOLUTION
PROGRAM RUN 3.208
RADIANS
2
4
6
8
10

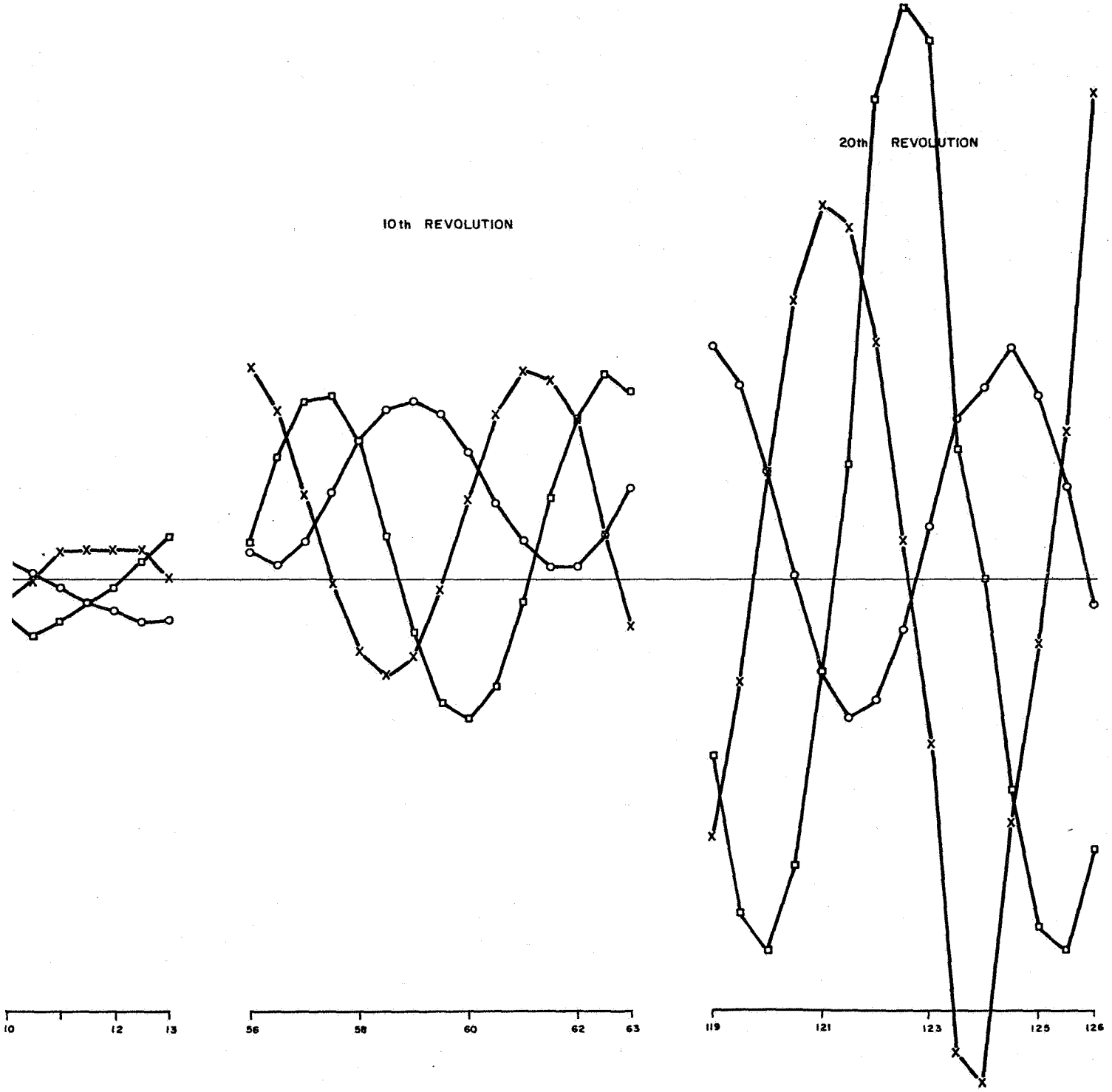
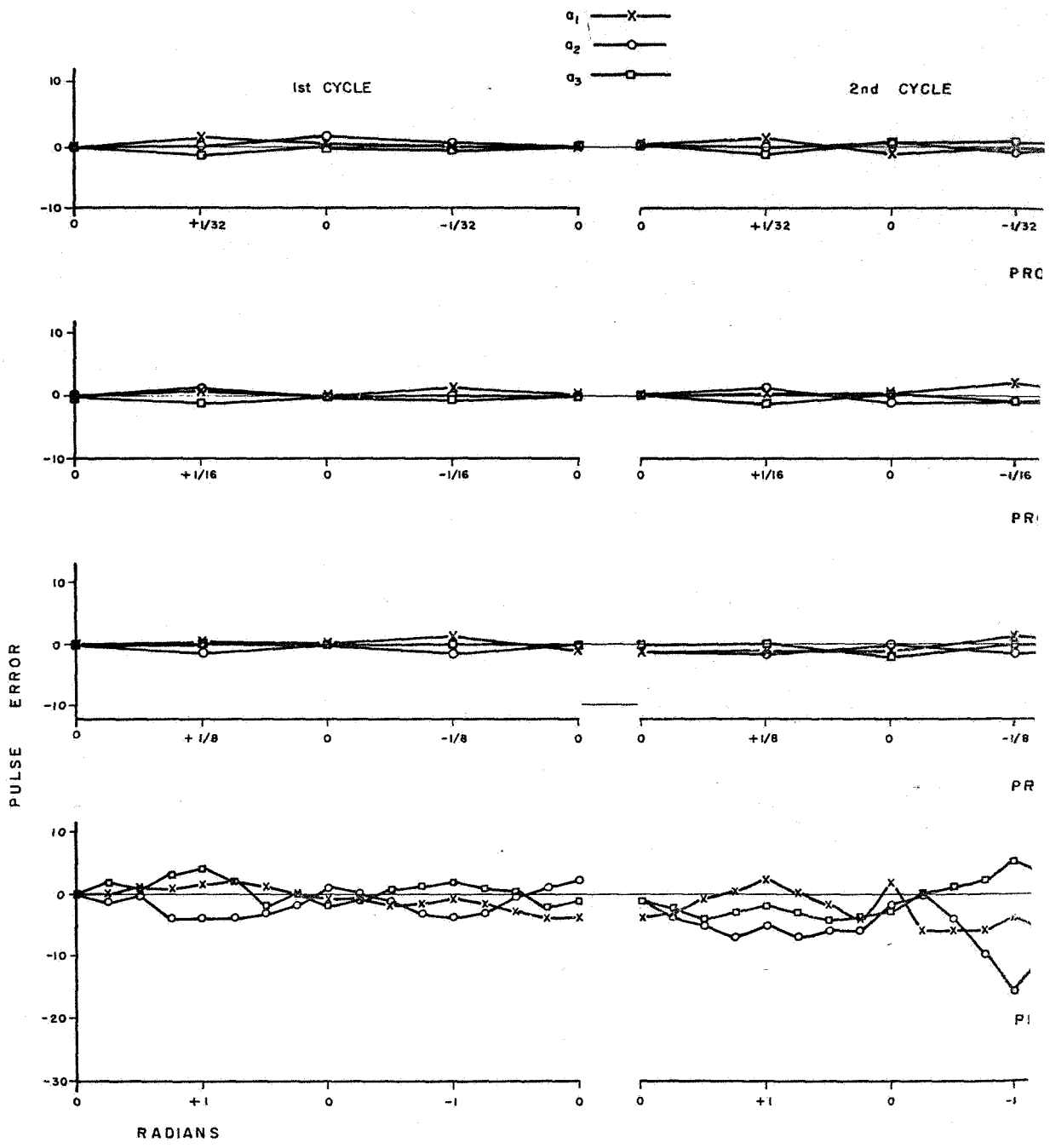

10th REVOLUTION
20th REVOLUTION
10
12
13
56
58
60
62
63
119
121
123
125
126

Figure B.6


a_1
a_2
a_3
1st CYCLE
2nd CYCLE
10
0
-10
0
+1/32
-1/32
+1/16
-1/16
+1/8
-1/8
PULSE ERROR
-20
-30
+1
-1
RADIANS

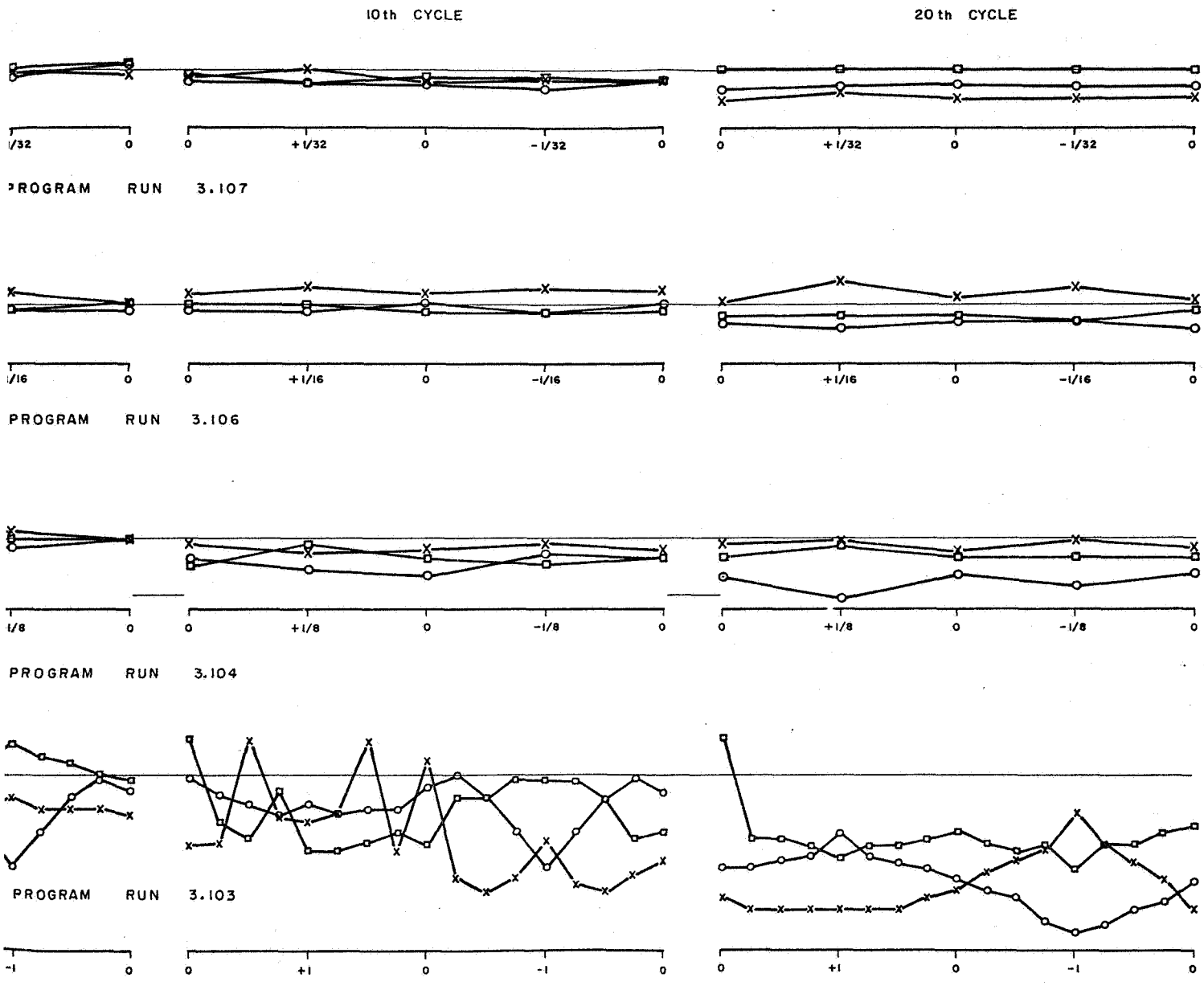
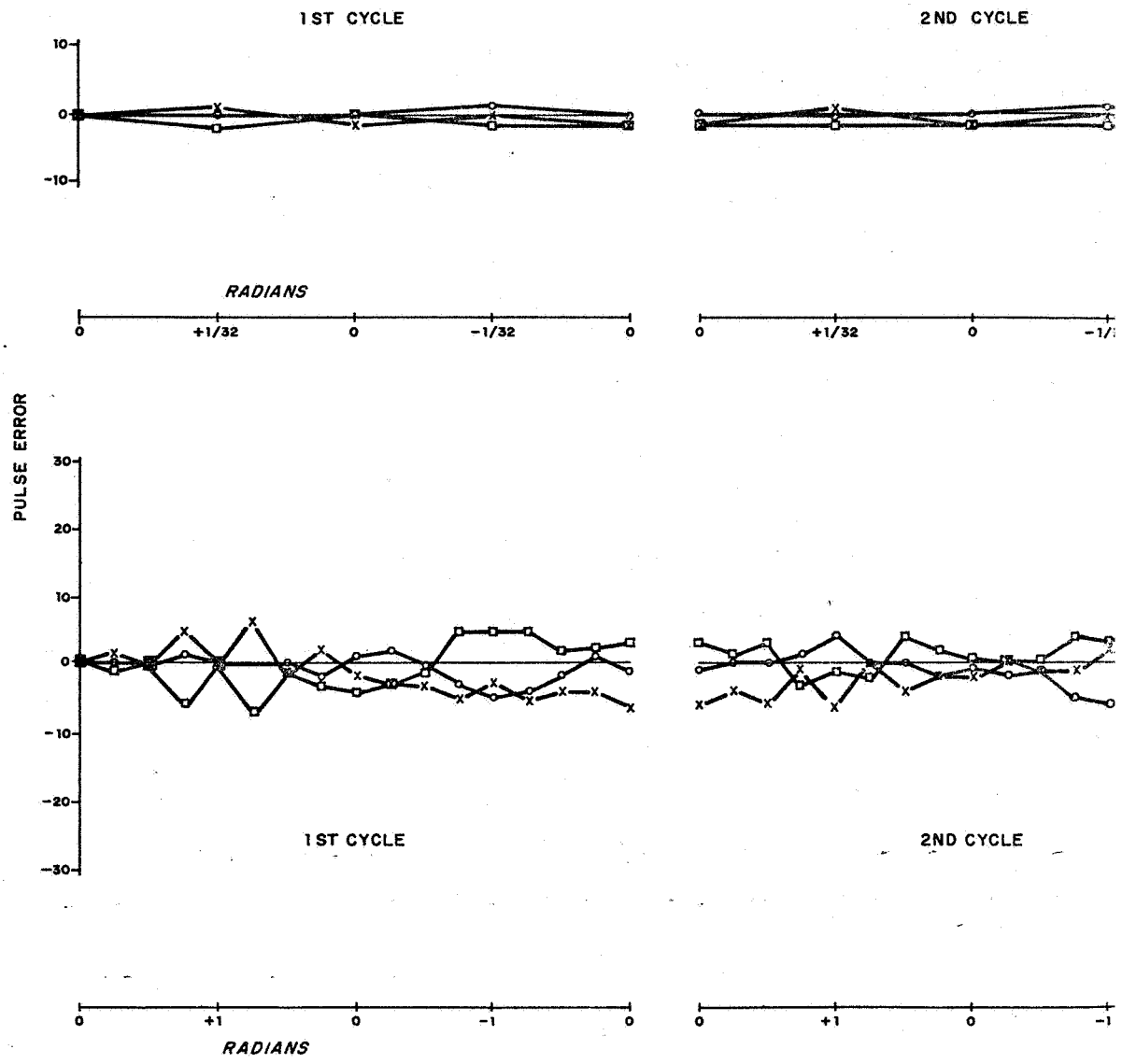
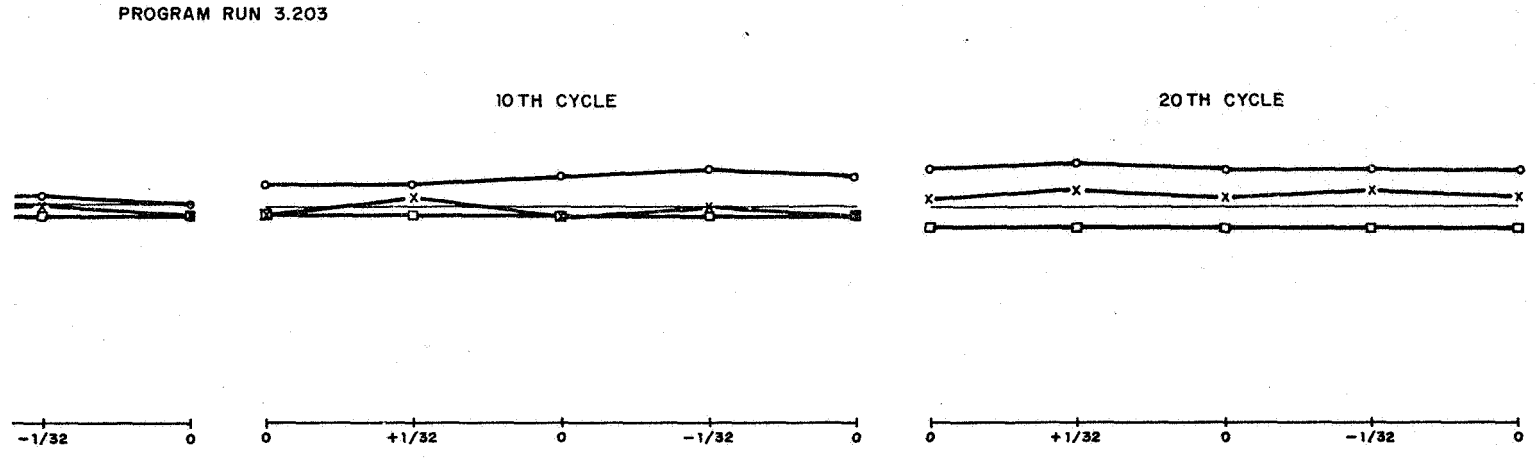

10th CYCLE
20th CYCLE
PROGRAM RUN 3.107
PROGRAM RUN 3.106
PROGRAM RUN 3.104
PROGRAM RUN 3.103

Figure B.7


1ST CYCLE
2ND CYCLE
10
0
-10
RADIANS
0
+1/32
0
-1/32
0
PULSE ERROR
30
20
10
0
-10
-20
-30
1ST CYCLE
2ND CYCLE
0
+1
0
-1
0
RADIANS


PROGRAM RUN 3.203
10 TH CYCLE
20 TH CYCLE
-1/32
0
0
+1/32
0
-1/32
0
0
+1/32
0
-1/32
0

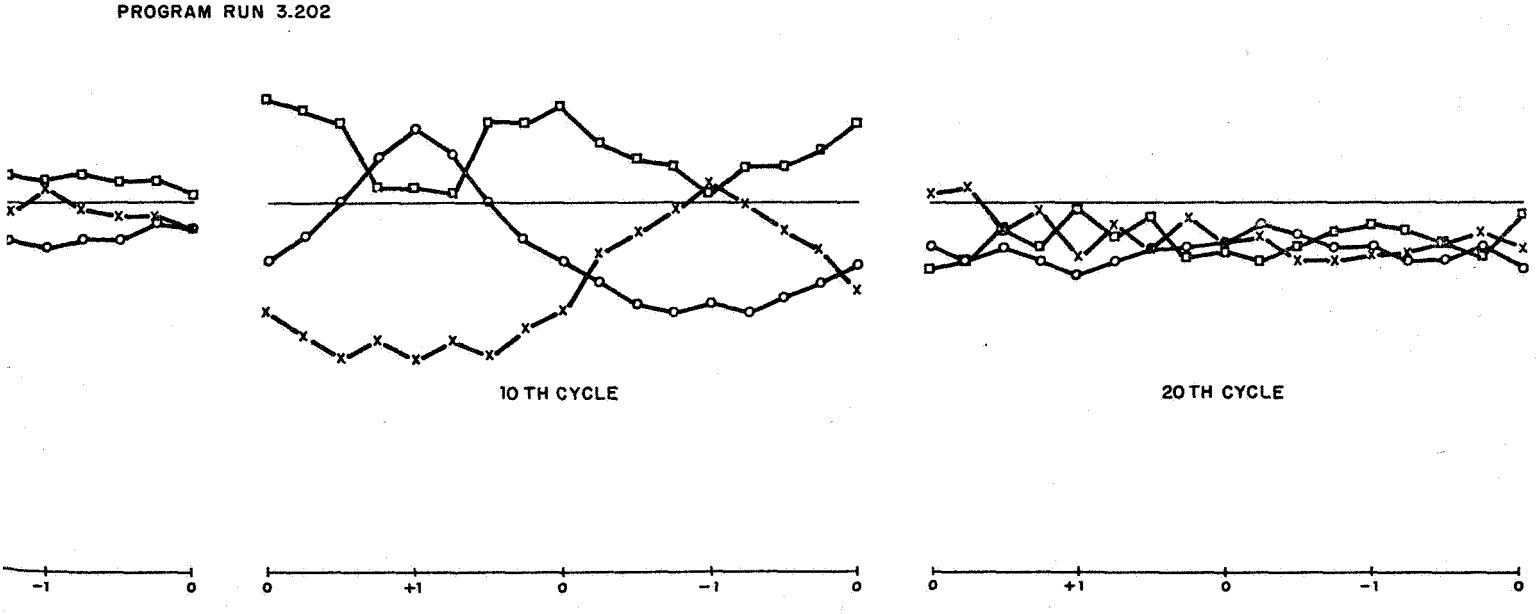

PROGRAM RUN 3.202
10 TH CYCLE
20 TH CYCLE
-1
0
0
+1
0
-1
0
0
+1
0
-1
0

Figure B.8

B.4 PROGRAM SET 4.000 – VELOCITY INVESTIGATION

B.4.1 Initial Conditions

Program	a_1	a_2	a_3	V_x^I
4.100 (Same as 3.100)	.5353 6860	-.1587 8187	.8295 5939	0
4.200 (Same as 1.100)	1.0		0.0	0
4.300 (Same as 1.200)	.8314 9663		.5555 2980	0

B.4.2 Program Runs

A. Ascent Program

	$\Delta\theta_i$ (Radians)	ΔV_x^B Relative Rate
4.101	$\Delta\theta_2$ CW(+), 0-1.5	$\Delta V_x^B = \frac{\Delta\theta_2}{16}$; $0 < \Delta\theta_2 \leq .5$
		ΔV_x^B (A.T.) = 102.4 ft/sec.
		$\Delta V_x^B = \frac{\Delta\theta_2}{4}$; $.5 < \Delta\theta_2 \leq 1.0$
		ΔV_x^B (A.T.) = 512 ft/sec.
		$\Delta V_x = \Delta\theta_2$; $1.0 < \Delta\theta_2 \leq 1.5$
		ΔV_x^B (A.T.) = 2150.4 ft/sec.
4.102	$\Delta\theta_2$ CW(+), 0-1.5	Same as 4.101
	$\Delta\theta_3$ Limit cycle ± 1/64	$\Delta\theta_3 = \frac{\Delta\theta_2}{8}$; $0 < \Delta\theta_2 \leq 1.5$

(A.T.) = Accumlative TOTAL increments of velocity at the end of each interval.

4.103 Same as 4.102, except limit cycle $\Delta\theta_1$ instead of $\Delta\theta_3$.

4.104 Same as 4.102, except limit cycle $\Delta\theta_1$ and $\Delta\theta_3$ in phase.

4.201 Limit cycle ΔV_x^B = 3276.8 ft./sec. for 20 cycles.

4.301 Same as 4.201.

B.4.3 Table of Results for Maximum Pulse Errors

A. Ascent Program

Table B.7

Maximum Pulse Errors					
Program Run	4.101	4.102	4.103	4.104	ΔV_x^B Slope
V_x^I	2	0	2	0	$\Delta\theta_2/16$; ΔV_x^B = 102.4 ft./sec.
	2	1	2	2	$\Delta\theta_2/4$; ΔV_x^B = 409.6 ft./sec.
	3	2	2	1	$\Delta\theta_2$; ΔV_x^B = 1638.4 ft./sec.
					ΔV_x^B (A.T.) = 2150.4 ft./sec.

B. Limit Cycle

Table B.8

	Maximum Pulse Errors		Maximum Pulse Errors	
Program Run	V_x^I	Program Run	V_x^I	Cycle
4.201	1	4.301	1	1
	1		1	2
	1		1	10
	1		1	20

B.5 ANALYTIC EXPRESSIONS PROGRAMMED FOR THE IBM 1401

We wish to obtain solutions to the three simultaneous differential equations,

$$\begin{aligned} da_1 &= a_2 d\theta_3 - a_3 d\theta_2 \\ da_2 &= a_3 d\theta_1 - a_1 d\theta_3 \\ da_3 &= a_1 d\theta_2 - a_2 d\theta_1 \end{aligned} \tag{B-2}$$

which are valid to six significant figures. The three dependent variables are a_1, a_2, a_3 and if we introduce the independent variable t, the above equations become

$$\begin{aligned} \dot{a}_1 &= a_2\dot{\theta}_3 - a_3\dot{\theta}_2 \\ \dot{a}_2 &= a_3\dot{\theta}_1 - a_1\dot{\theta}_3 \\ \dot{a}_3 &= a_1\dot{\theta}_2 - a_2\dot{\theta}_1 \quad . \end{aligned} \tag{B-3}$$

In general these equations can be integrated only by numerical methods but for the special case when the angles θ_i are changing at a constant rate

$$\dot{\theta}_i = k_i \quad . \tag{B-4}$$

Equations (B-3) can be conveniently written in the vector form

$$\dot{\bar{a}} = \bar{a} \times \bar{k} \quad . \tag{B-5}$$

The fact that $\dot{\bar{a}}$ is perpendicular to $\bar{a}$ implies that $\bar{a}$ does not change in magnitude, and the fact that it is also perpendicular to the fixed vector $\bar{k}$ means that $\bar{a}$ generates a cone about an axis defined by $\bar{k}$. From the law of cosines for oblique spherical triangles and Figure B.9, we obtain

$$a_1 = \cos b \cos c - \sin b \sin c \cos \alpha_1 \tag{B-6}$$

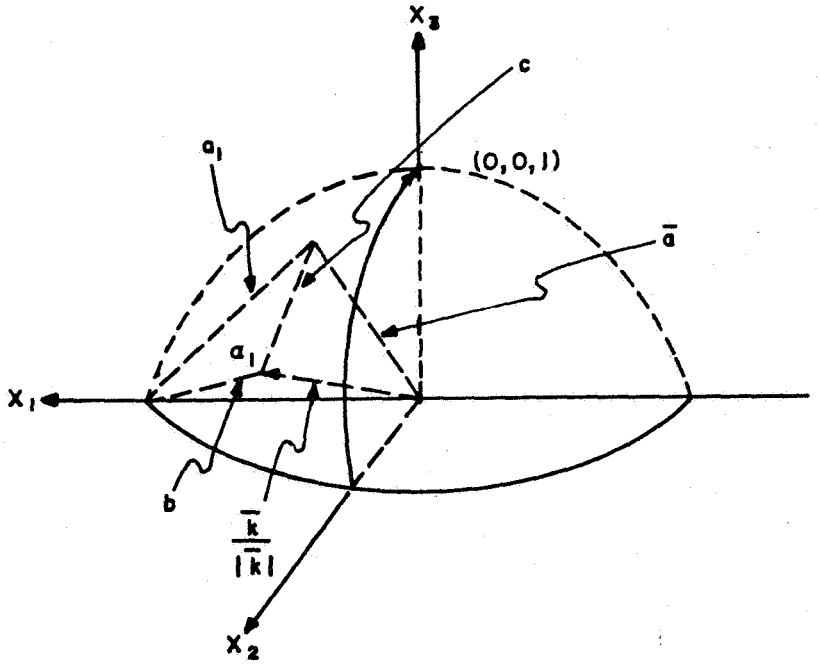


Figure B.9 Direction-Cosine Vector Diagram

where $\cos b = \frac{k_1}{|\bar{k}|}$, $\cos c = (a_{10}k_1 + a_{20}k_2 + a_{30}k_3)\ \frac{1}{|\bar{k}|}$, $\alpha_1 = |\bar{k}|t + \alpha_{01}$;

a_{10}, a_{20}, a_{30}, α_{01} are the given initial values for $t = 0$.

Similarly, formulas are readily obtained for a_2 and a_3:

$$a_i = \cos b \cos c + \sin b_i \sin c \cos (\omega t + \alpha_{oi}) \quad (i = 1, 2, 3)$$

$$= C_i + D_i \cos (\omega t + \alpha_{oi})$$

where

$$\cos b_i = \frac{k_i}{|\bar{k}|}$$

$$\cos c = \bar{a}_o \cdot \frac{\bar{k}}{|\bar{k}|} \qquad \text{(B-7)}$$

$$\omega = |\bar{k}| \quad .$$

For the cases considered in this report, after the constants C_i, D_i, and α_{oi} were evaluated, the resulting formulas for a_i were checked by substituting numerically into (B-3) or solutions were additionally obtained (at least over a limited range) by a numerical method. This later check was made primarily to study the accuracy of the numerical method.

The numerical solution of (B-3) that was selected and programmed is known as Hamming's method. A detailed discussion of this method can be found in "Ralston and Wolf, Mathematical Methods for Digital Computers, John Wiley & Sons, 1960." It was found that by using nine-significant-digit floating point accuracy and an integration interval, Δt, of about 0.03 or 0.04 errors could be held below one or two in the 6th decimal point in the range $0 \leq |t| \leq 20\,(2\pi)$, for those cases that could be checked analytically. The errors oscillate and are usually considerably less than this.

An additional task was the evaluation of the expression

$$V_X^I = \int a_1 dV_X^B \qquad \text{(B-8)}$$

where a_1 is as defined in (B-7) and

$$V_X^B = p(t) \cdot t \qquad \text{(B-9)}$$

where p (t) is a piecewise constant changing only at given values of t. Substituting (B-7) and (B-9) into (B-8) we obtain (over an interval of constant p)

$$V_X^I = p \int_0^{t_1} [C_1 + D_1 \cos(\omega t + \alpha_{oi})]dt$$

$$= pC_1 t_1 + p \frac{D_1}{\omega} [\sin(\omega t_1 + \alpha_{oi}) - \sin \alpha_{oi}] \quad .$$

Since for the cases considered p, and k_i were all piecewise constants this formula was used repeatedly, it being necessary of course to re-evaluate the constants C_1, D_1, ω, α_{oi} at the start of each interval.

The evaluation of the analytic expressions was programmed in the IBM 1401 Fortran using 9-significant digit accuracy.

APPENDIX C

AIRBORNE COMPUTER ANALYSIS

C.1 Description of T-1 Computer

Figure C.1 shows the basic arrangement and signal flow for the forward-backward counter. For the reference application at 2^{-13} radian, the counter must be 16 bits long to cover smoothing and polarity requirements. Each such counter is designed as two identical stacks, ganged together to utilize a uniform building-block size appropriate for mechanical design, reliability and maintainability purposes.

The sub-multiplier shown on Figure C.2 performs two-thirds of the pulse multiplier function, gating the magnitude part of the code input with the input pulse rate or increment. The partial product is then converted to a full product in the product section of the product adder shown in Figure C.3. Here the "two's" complement of the partial product is obtained for negative codes, and product polarity is determined. The same sub-assembly contains an uncommitted pulse adder with two inputs and two stages of smoothing.

Figure C.4 shows the arrangement for obtaining synchronously timed pulses in accordance with the external input signals. This is followed by the upscaler of Figure C.5, which quadruples the pulse quantity in order to compensate for smoothing in the counters. The clock slot generator for the computer is shown in Figure C.6. Discrete circuits are used for the astable multivibrator and the output pulse amplifiers, and integrated circuits for the remainder.

Computer composition is tabulated by stack in Table C-1. Inspection of Figure 3.2 shows that only 15 counters, 27 multipliers and 15 two-input adders are necessary. However, three additional counters are needed due to a maximum rate fanout capability of three, which is inadequate for the angle counters. The uniform building block approach also provides 12 unused adders, which are inconvenient to remove; their effect on component count and power drain is only five percent.

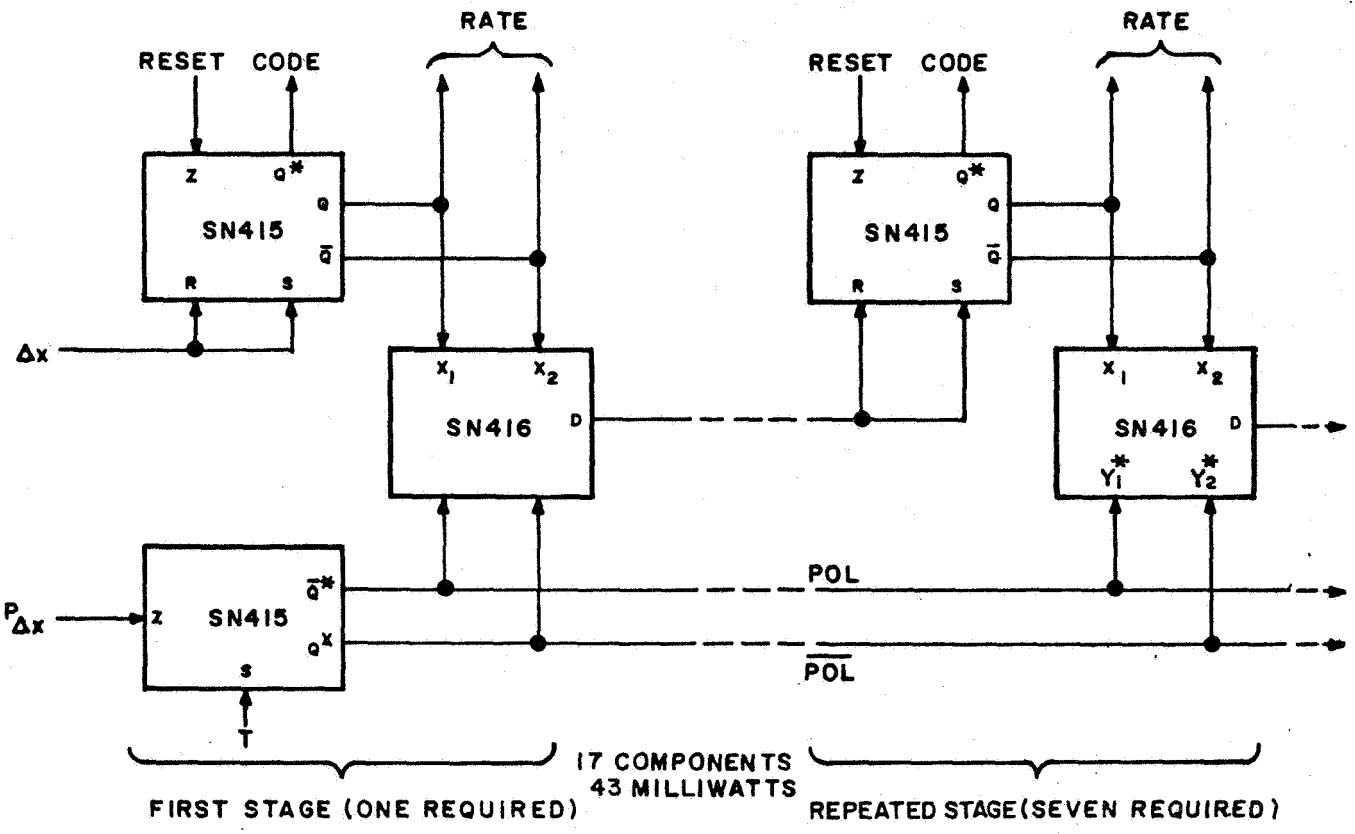


Figure C.1 Block Diagram of Counter FBC-8D for T-1 Computer

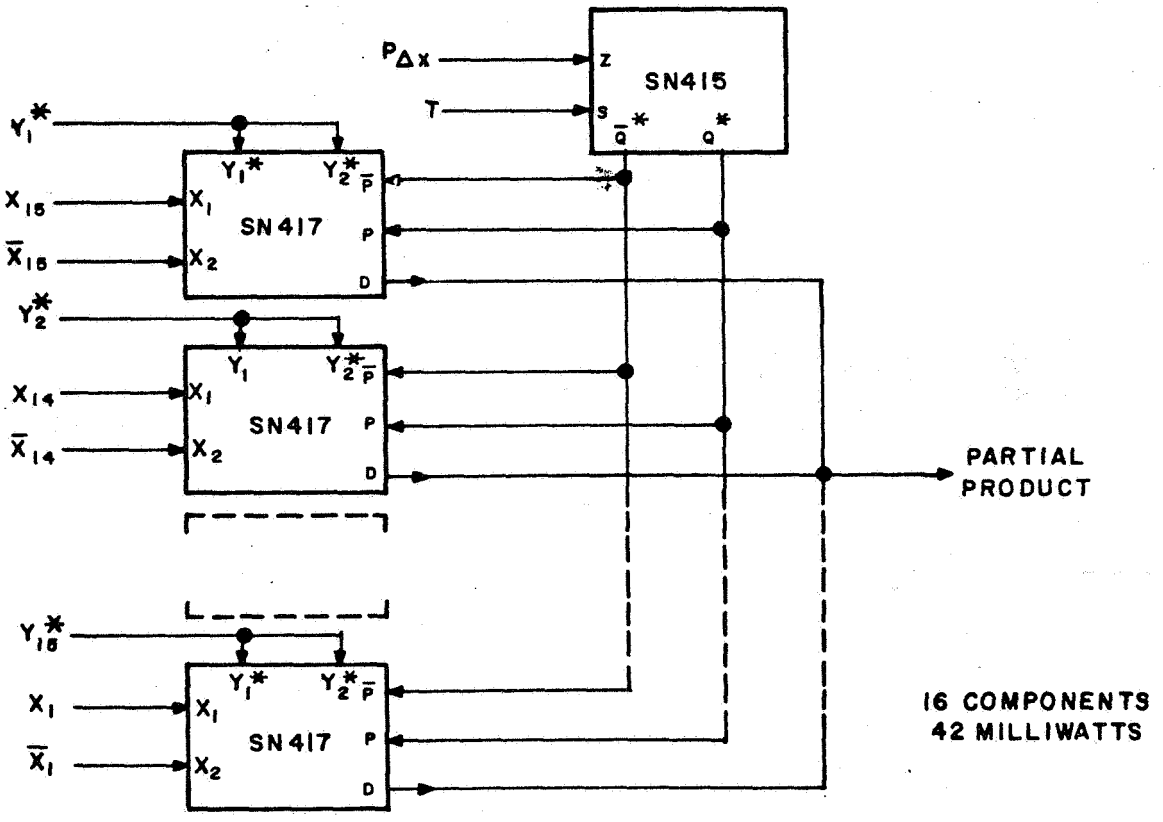


Figure C.2 Block Diagram of Sub-Multiplier MS-15 for T-1 Computer

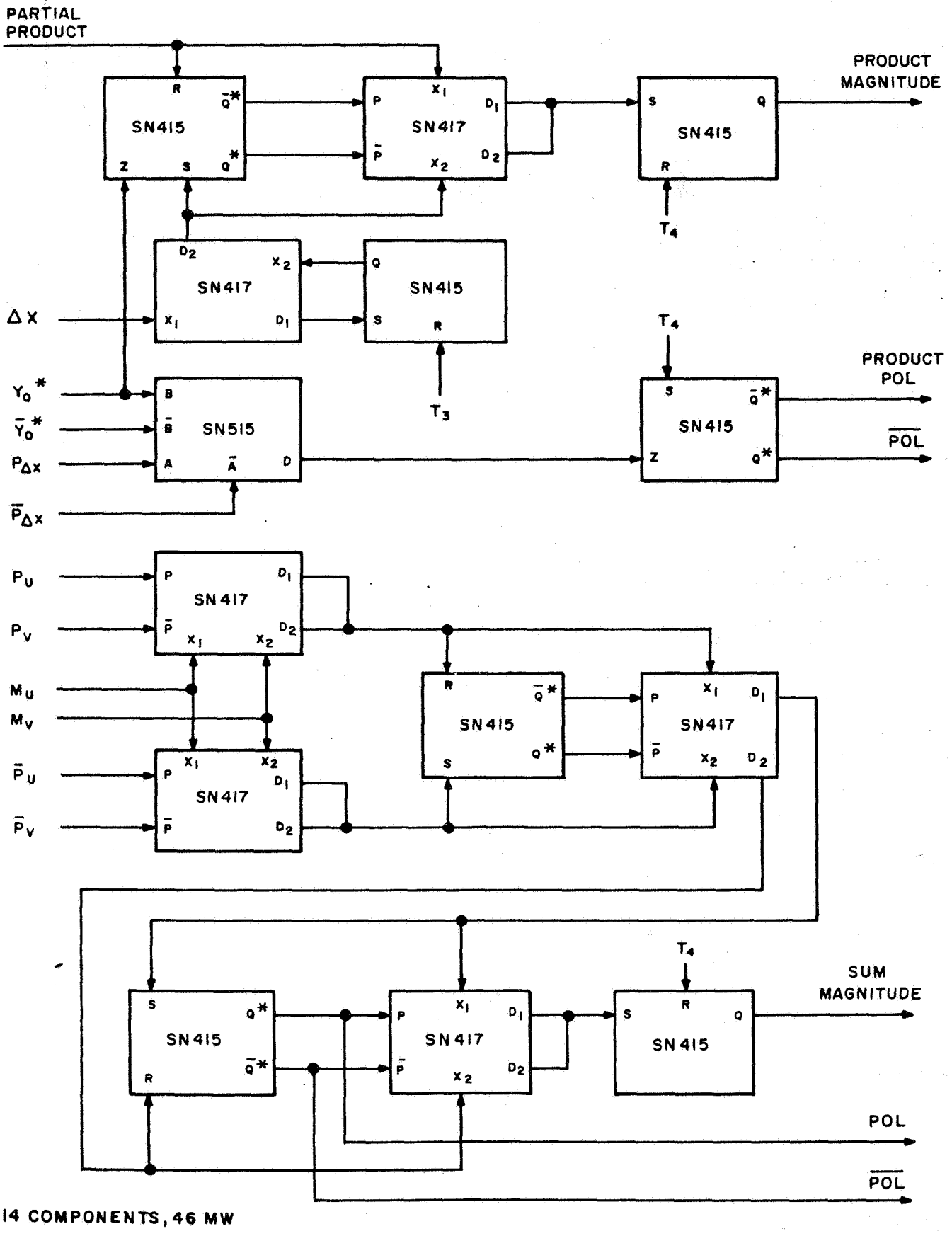
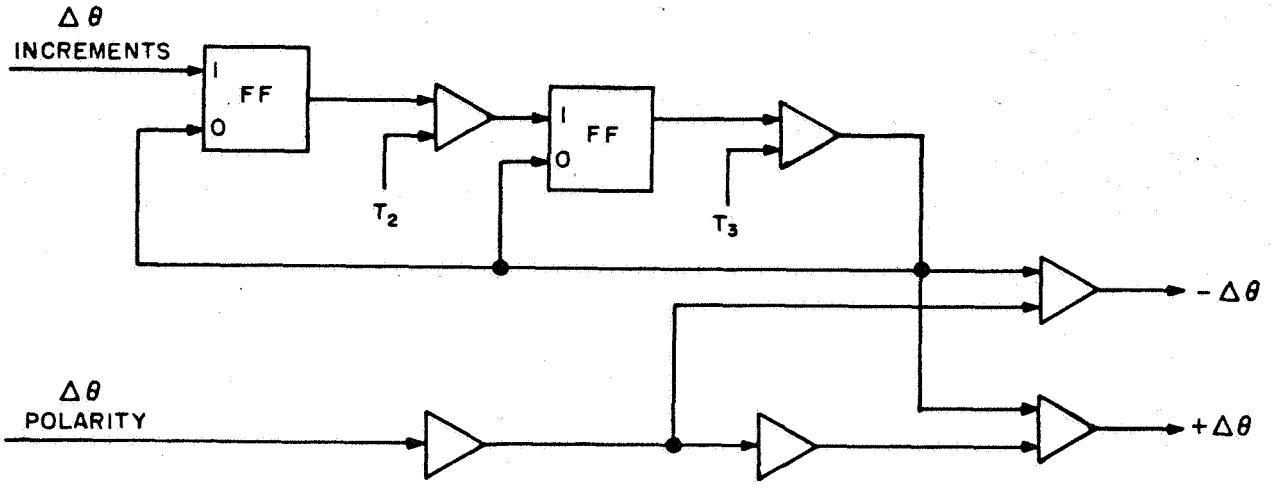
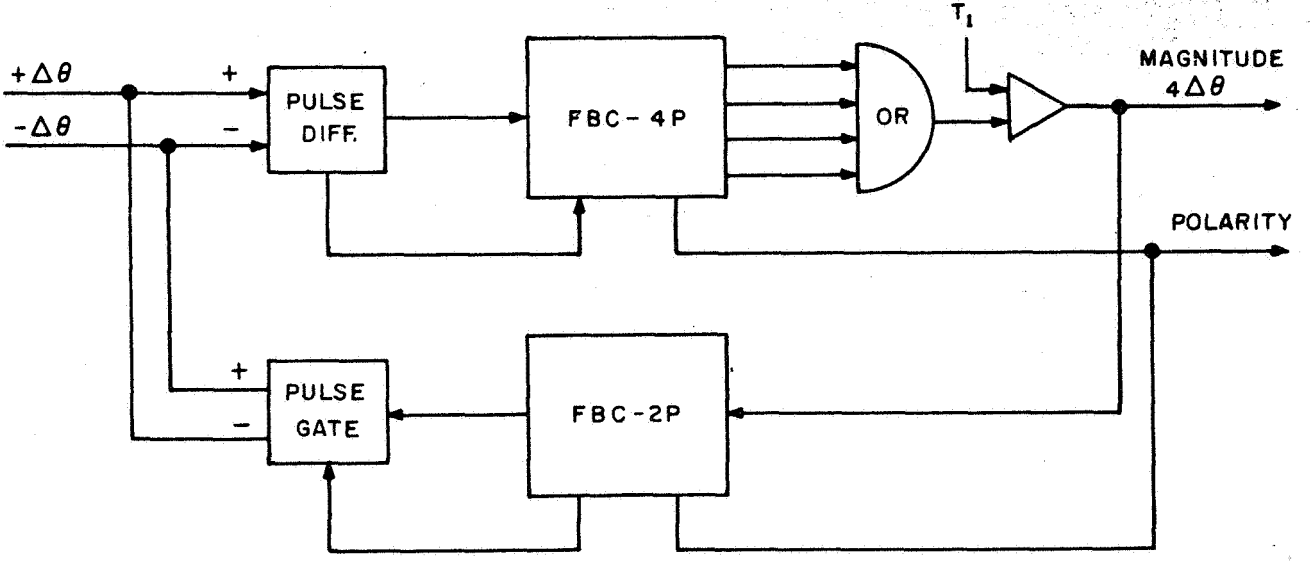


Figure C.3 Block Diagram of Product Adder MP/A for T-1 Computer



T-1 AND T-2 COMPUTERS: 3 EA SN415, AND 3 EA SN417; 20 MW
F-1 COMPUTER: 8 EA 910, AND 2 EA 913; 76 MW
F-2 COMPUTER: 8 EA 914, AND 2 EA 916; 300 MW

Figure C.4 Block Diagram of Synchronizer SI-3; Each of Three Channels



T-1 AND T-2 COMPUTERS: 10 EA SN415, 3 EA SN417, 1 EA SN514; 40MW
F-1 COMPUTER: 13 EA 910, 1 EA 911, 7 EA 912, 10 EA 913; 232 MW
F-2 COMPUTER: 7 EA 940, 1 EA 907, 4 EA 910, 9 EA 914, 10 EA 916; 1.10 W

Figure C.5 Block Diagram of Upscaler

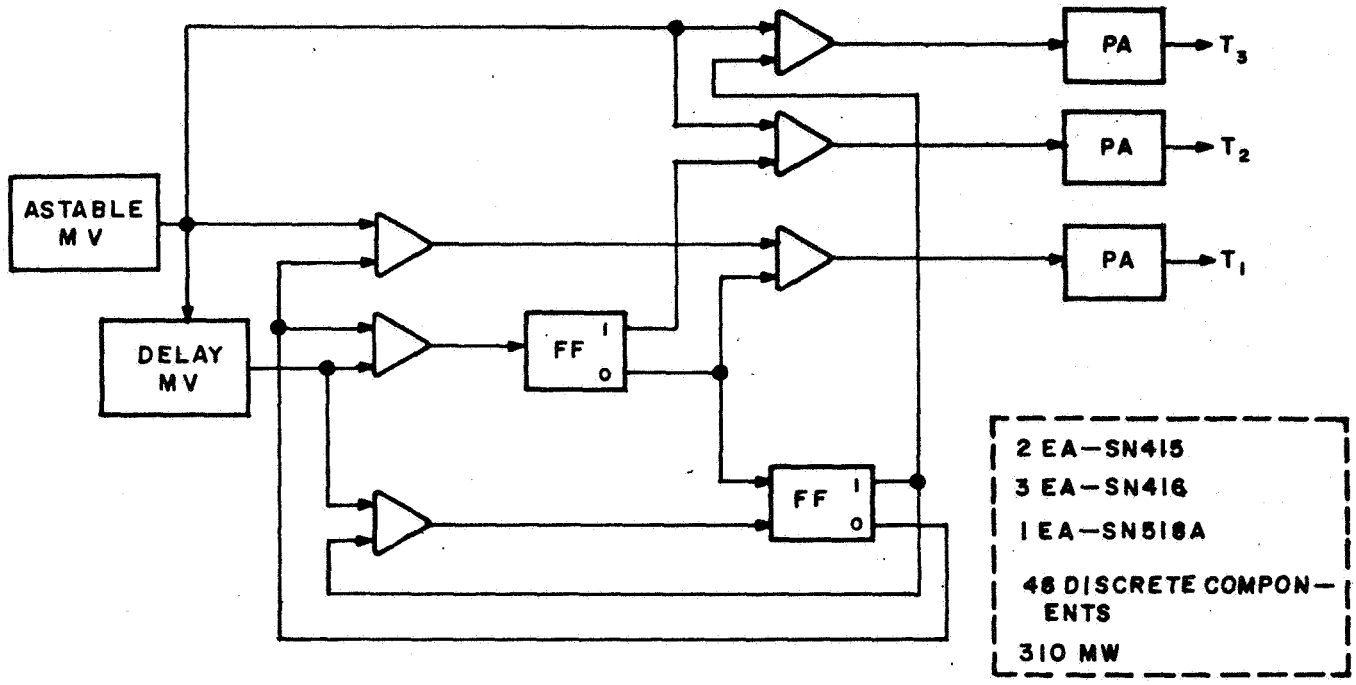


Figure C.6 Block Diagram of Clock Slot Generator for T-1 and T-2 Computers

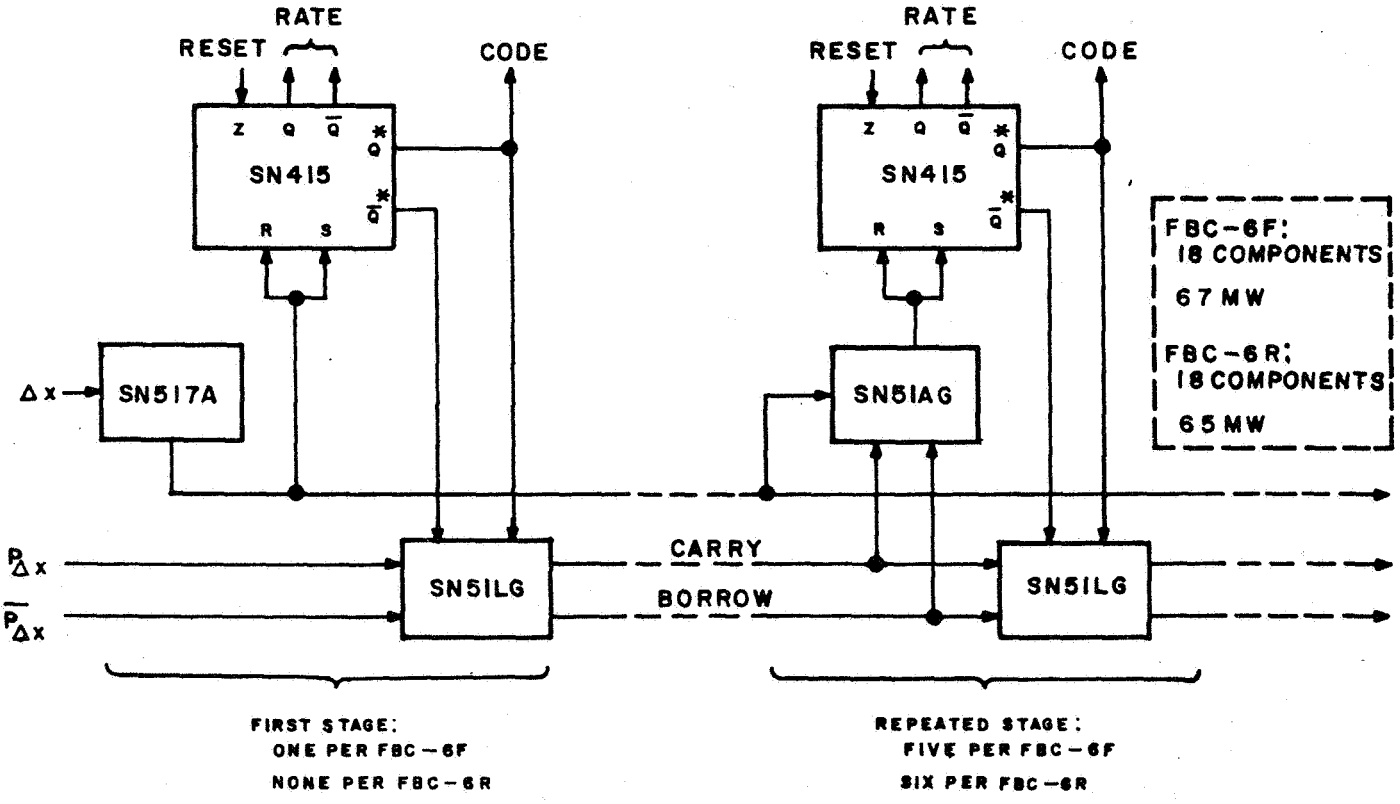


Figure C.7 Block Diagram of Counter FBC-6 for T-2 Computer

Table C-1. Composition of Computer T-1 by Stack

36	FBC-8D	at	17 Components,	43 MW	:	612 Components	1.55W
27	MS-15		16	42		432	1.13
27	MP/A		14	46		378	1.24
2	SI-3		18	60		36	0.12
6	Upscaler		14	40		84	0.24
2	Clock					54	0.31
100	Sub-assemblies					1596 Components	4.59W

C.2 Description of T-2 Computer

The T-2 Computer differs from the T-1 mainly in the replacement of the ripple-carry, forward-backward counter with the fast-carry design of Figure C.7. The required 18-bit word length is achieved by stacking three unit sub-assemblies. The remaining differences are the increased multiplier word length, and added gating for the synchronizers in the product multiplier in order to speed up synchronizer recovery time. Computer composition is summarized in Table C-2.

Table C-2. Composition of Computer T-2 by Stack

18	FBC-6F	at	18 Components,	67 MW	:	324 Components	1.21W
36	FBC-6R		18	65		648	2.34
27	MS-17		18	47		486	1.27
27	MP/A		16	54		432	1.46
2	SI-3		18	60		36	0.12
6	Upscaler		14	40		84	0.24
2	Clock					54	0.31
118	Sub-assemblies					2064 Components	6.95W

C.3 Description of F-1 Computer

The design for a forward-backward counter using Fairchild Milliwatt Micrologic is shown in Figure C.8. The first stage provides several outputs delivered to all of the succeeding stages in parallel. Each counter is implemented by cascading small uniform-size, five-bit sub-assemblies.

The sub-multiplier shown on Figure C.9 performs the same function as described for the T-1 Computer. The same applies to the product adder shown in Figure C.10.

The synchronizer and upscaler are covered in Figures C.4 and C.5. The clock slot generator is organized similarly to that shown in Figure C.6, except that cascaded integrated circuits are used for the output drivers and for the astable multivibrator.

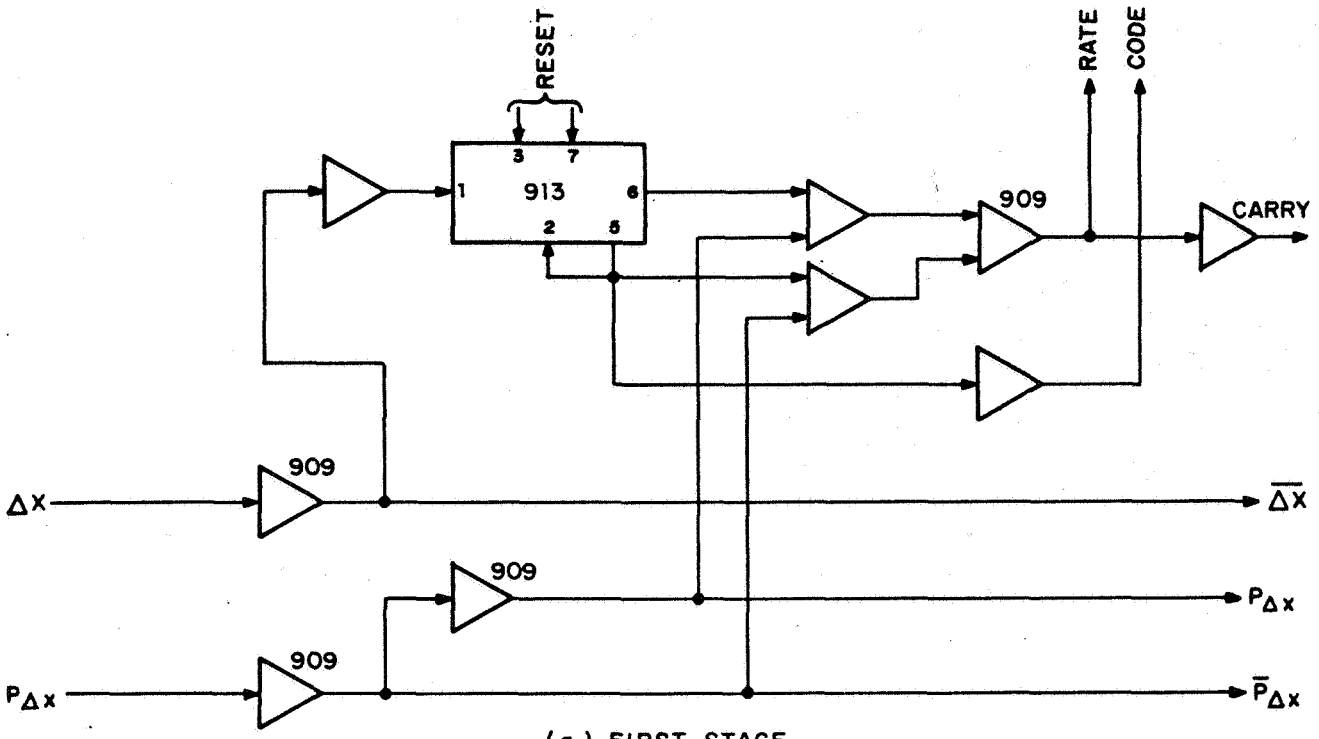
Table C-3 summarizes computer composition by stack.

Table C-3. Composition of Computer F-1 by Stack

15	FBC-5F	at	32	Components,	0.23 W	: 480	Components	3.45 W
45	FBC-5R		30		0.21	1350		9.45
27	MS-19		27		0.26	729		7.02
27	MP/A		30		0.17	810		4.59
2	SI-3		30		0.23	60		0.46
6	Upscaler		31		0.23	186		1.38
1	Clock		20		0.18	20		0.18
123	Sub-assemblies					3635	Components	26.53 W

C.4 Description of F-2 Computer

The F-2 Computer is a standard Micrologic extension of the F-1 design, as may be seen from Figures C.11, C.12 and C.13. Computer composition by stack is summarized in Table C-4.



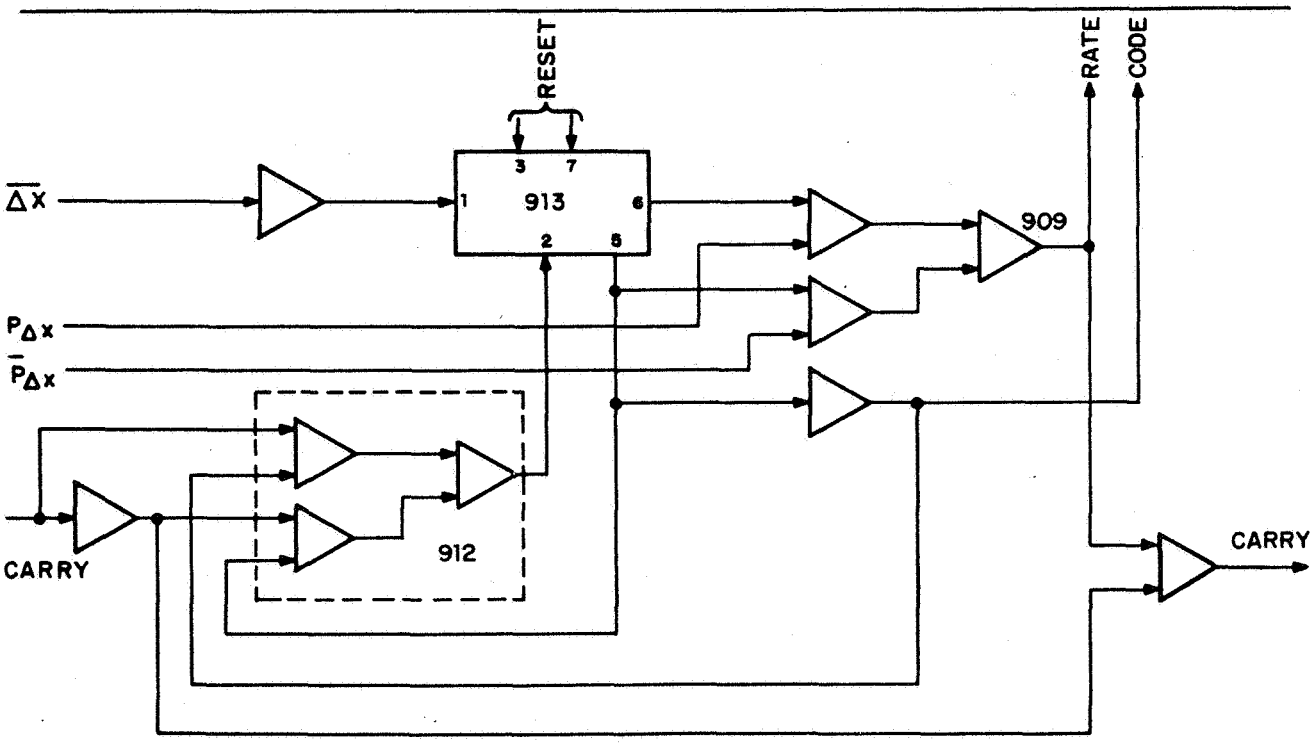
(a) FIRST STAGE

8 COMPONENTS, 64 MW

UNLABELLED COMPONENTS 1/2 x 910

ONE PER FBC-5F

NONE PER FBC-5R



(b) REPEATED STAGE

6 COMPONENTS, 42 MW

4 PER FBC-5F

5 PER FBC-5R

Figure C.8 Block Diagram of Counter Stages for F-1 Computer

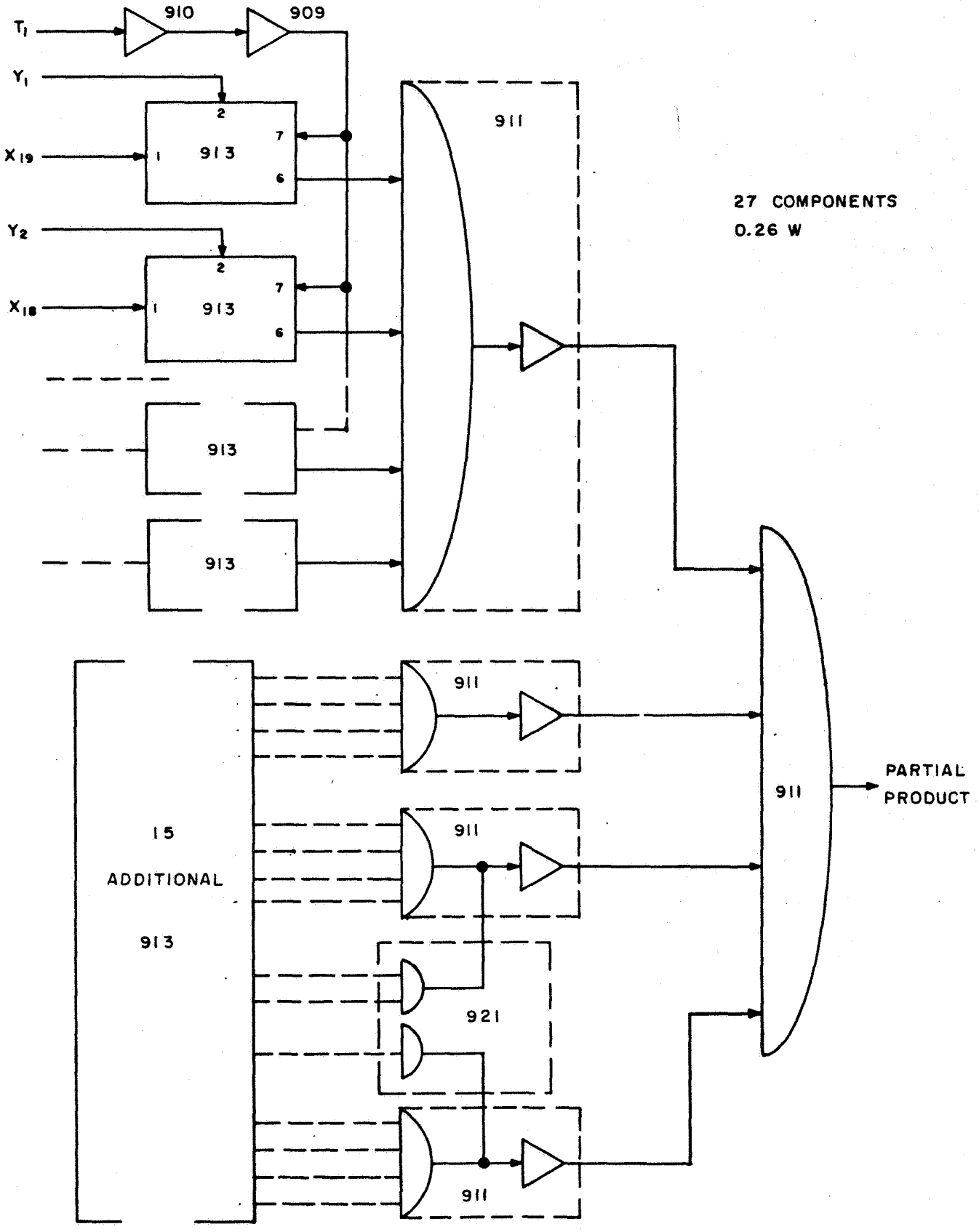


Figure C.9 Block Diagram of Sub-Multiplier MS-19 for F-1 Computer

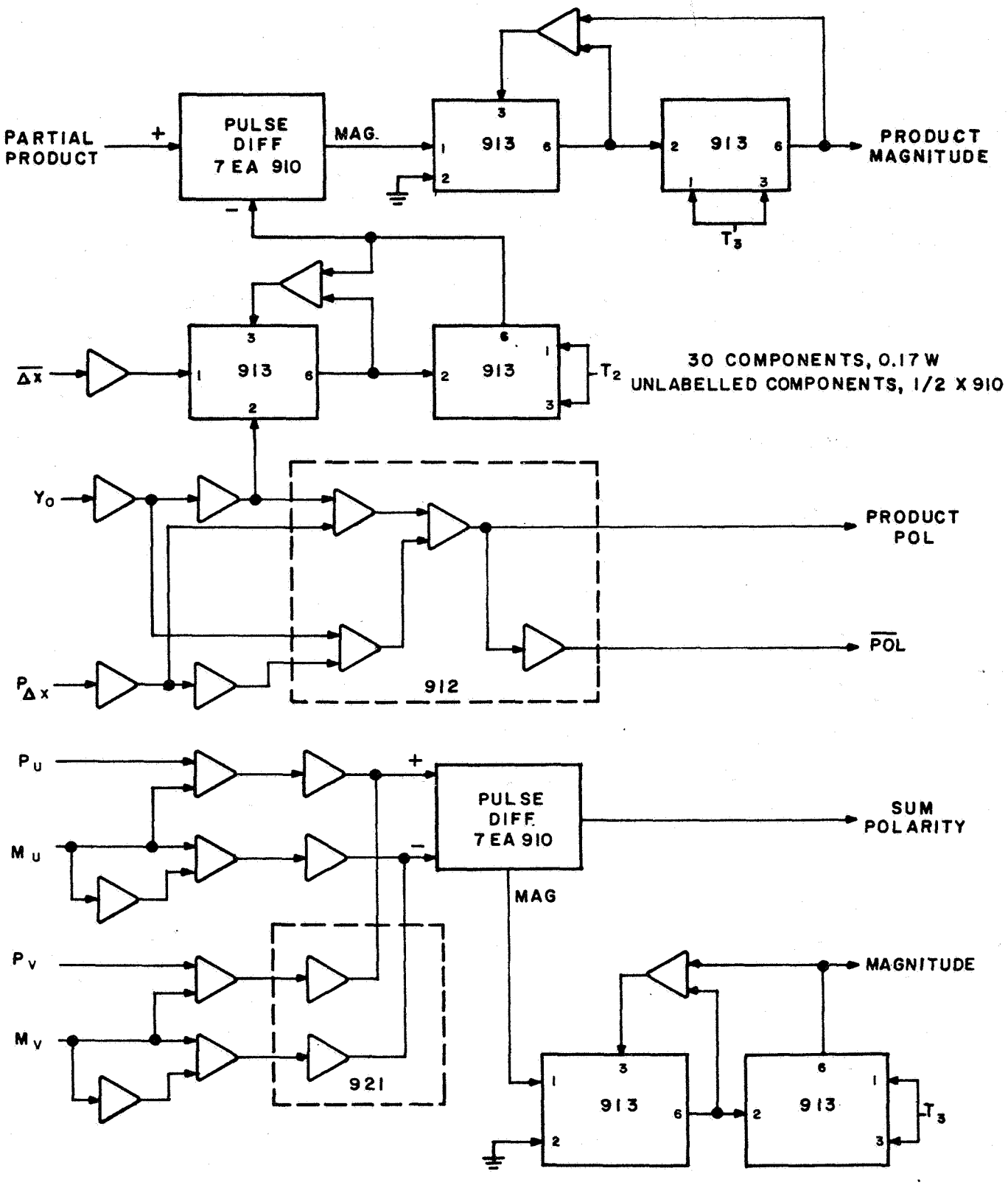
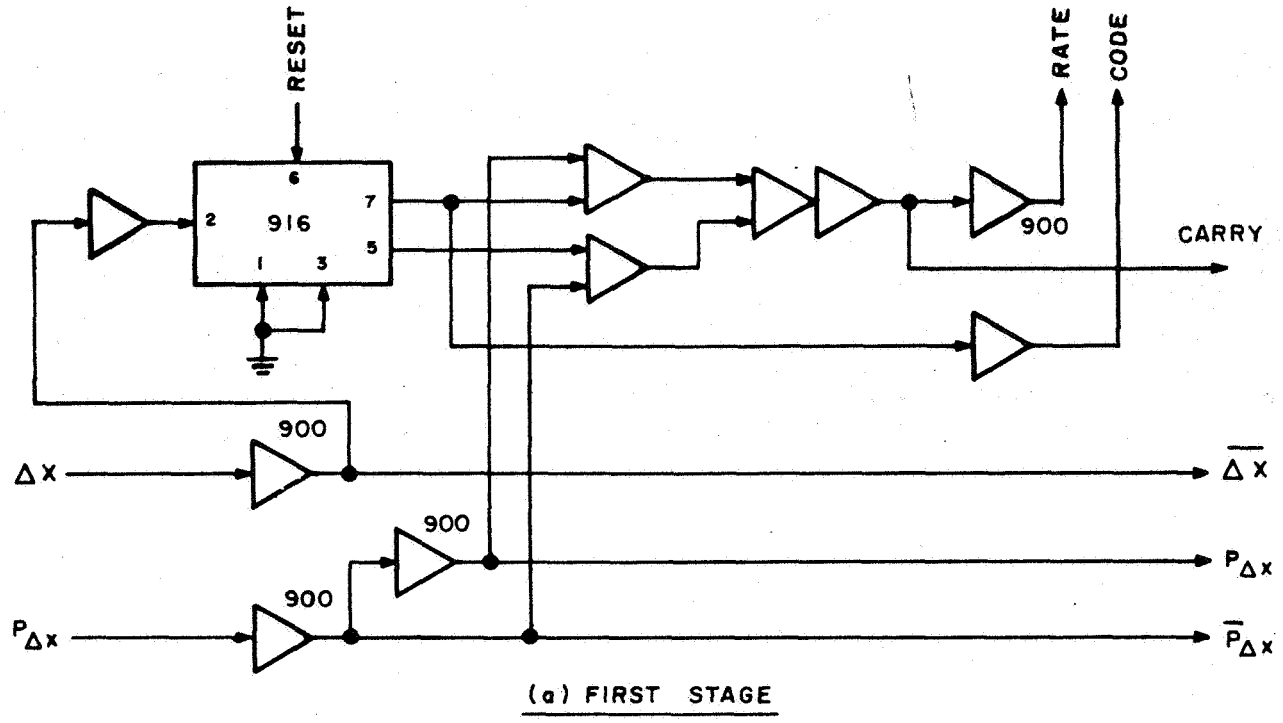


Figure C.10 Block Diagram of Product Adder MP/A for F-1 Computer



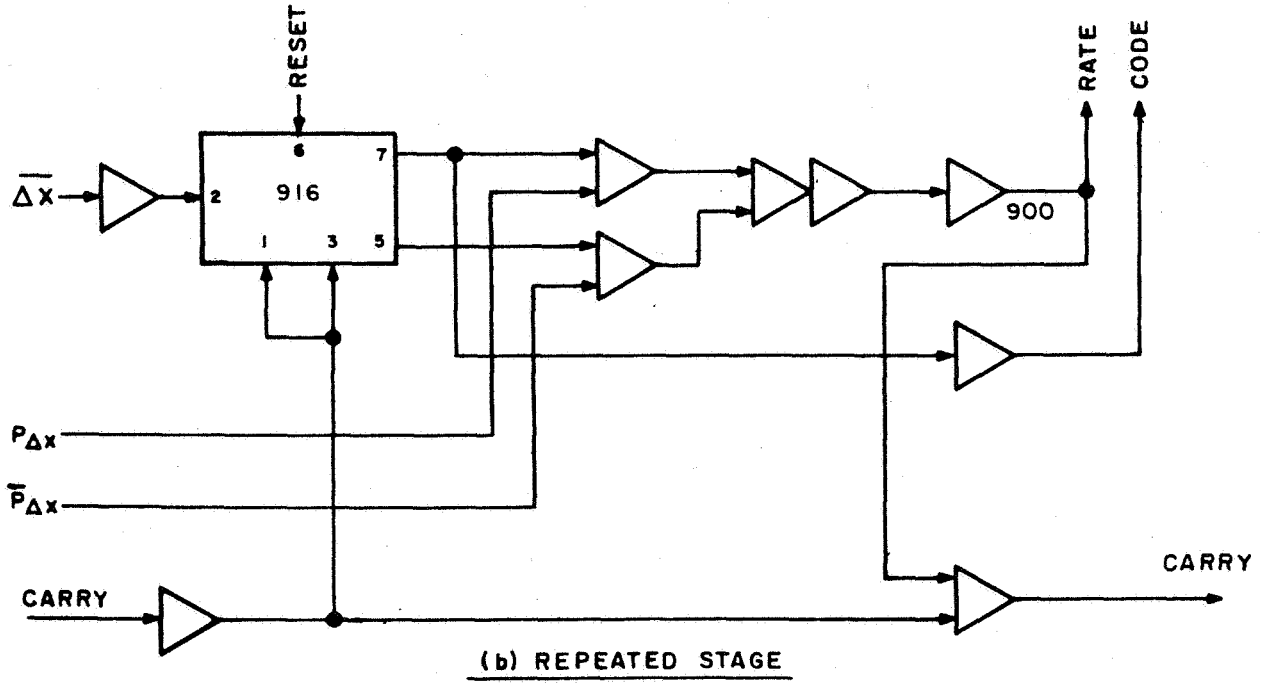
(a) FIRST STAGE

8 COMPONENTS, 0.26 W

UNLABELLED COMPONENTS 1/2 x 914

ONE PER FBC-4F

NONE ELSEWHERE



(b) REPEATED STAGE

6 COMPONENTS, 0.16 W

UNLABELLED COMPONENTS 1/2 x 914

3 PER FBC-4F

5 PER FBC-5R

4 PER FBC-2D

Figure C.11 Block Diagram of Counter Stages for F-2 Computer

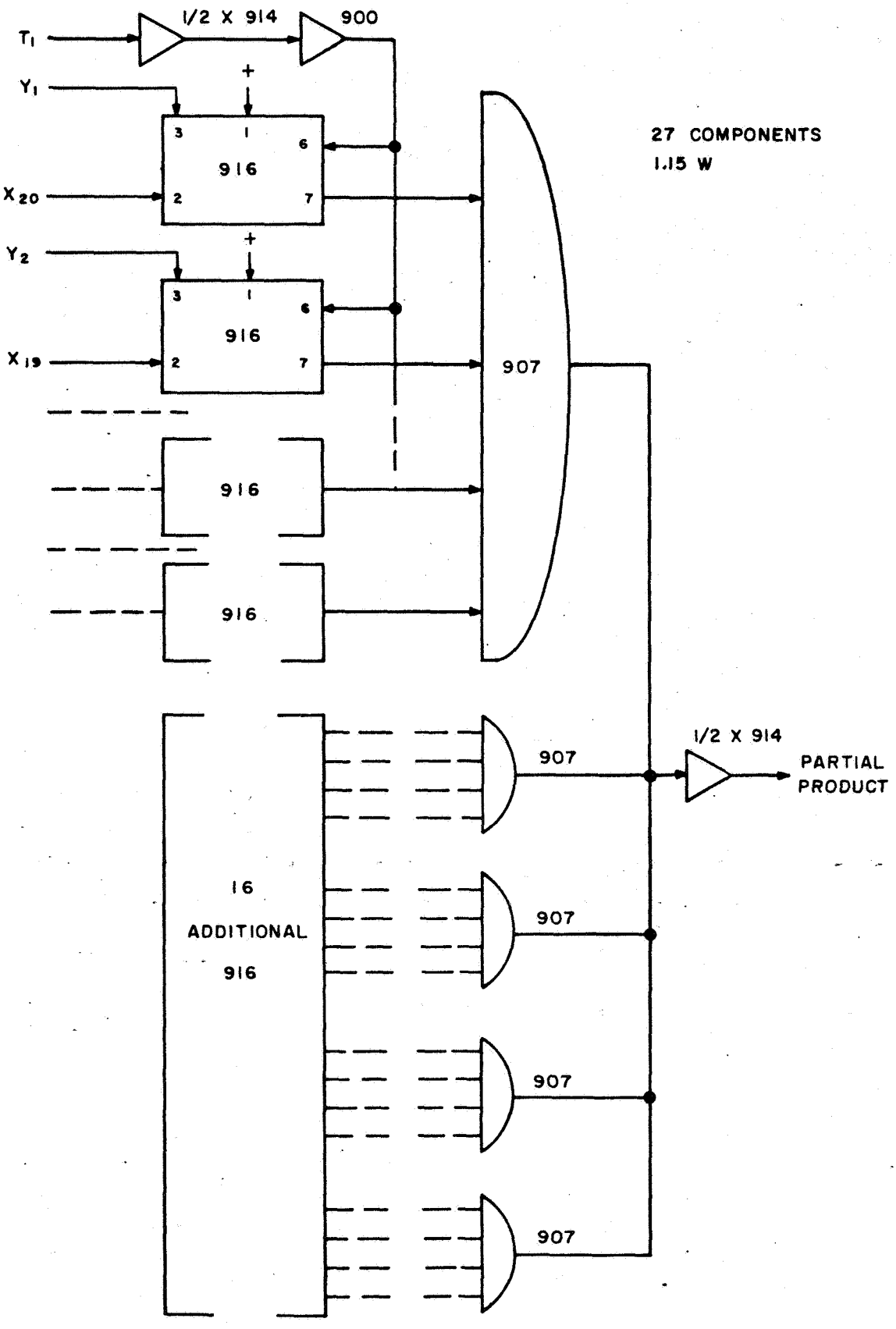


Figure C.12 Block Diagram of Sub-Multiplier MS-20 for F-2 Computer

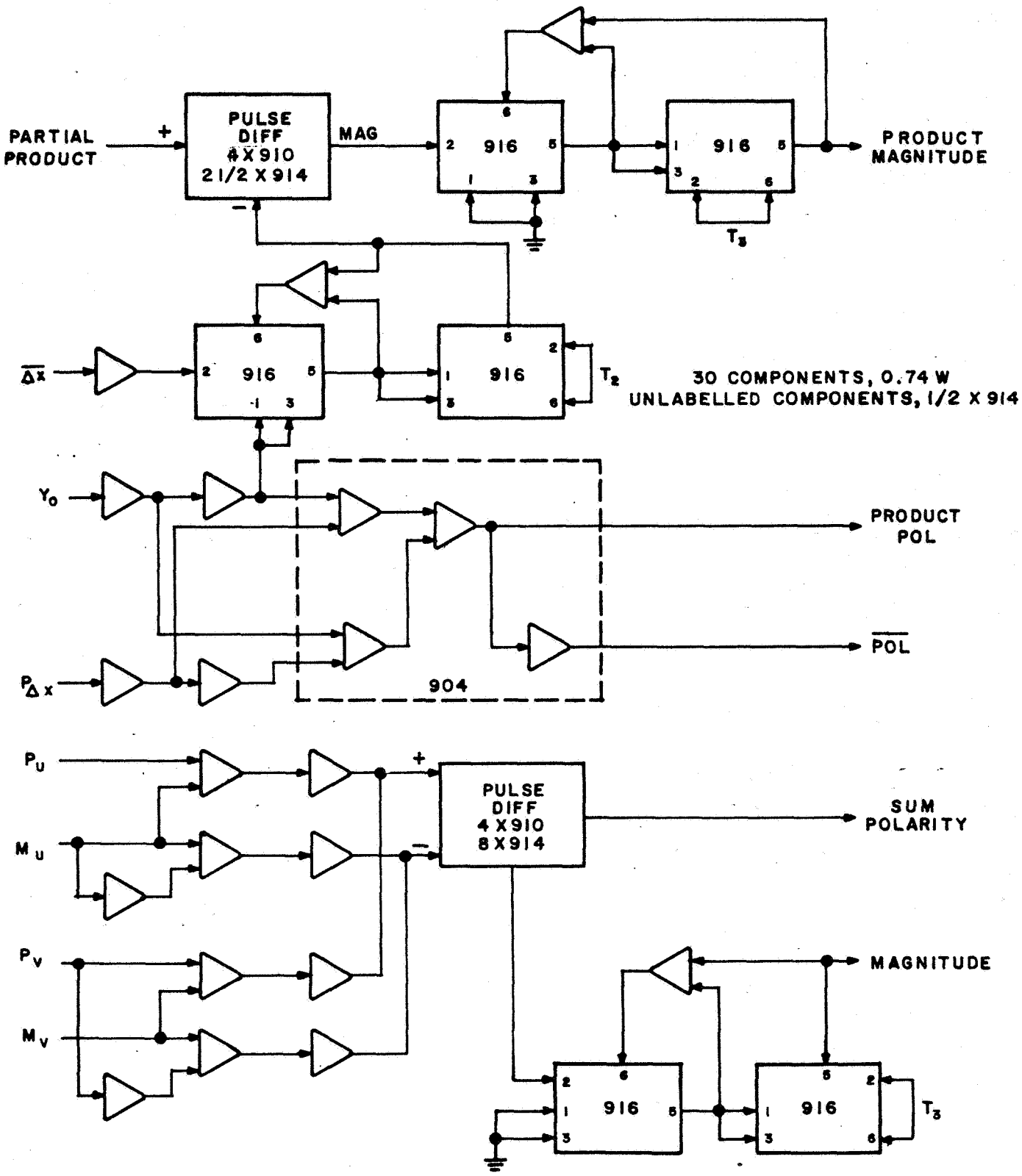


Figure C.13 Block Diagram of Product Adder MP/A for F-2 Computer

Table C-4. Composition of Computer F-2 by Stack

15	FBC-4F	at	26	Components,	0.74 W	: 390	Components	11.1 W
45	FBC-5R		30		0.80	1350		36.0
8	FBC-2D		24		0.64	192		5.1
27	MS-20		27		1.15	729		31.1
27	MP/A		30		0.74	810		20.0
2	SI-3		30		0.90	60		1.8
6	Upscaler		31		1.10	186		6.6
2	Clock					33		0.6
132	Sub-assemblies					3750	Components	112.3 W

C.5 Power Supply

The power supply unit is designed to produce output voltages of +6V DC and +3V DC at four different power levels from an incoming 28V DC supply. It has been assumed that the incoming supply will vary between the limits of 25V and 31V, and that large voltage transients will occasionally be present. A summary of the power supply unit characteristics is presented in Table C-5. Due to the method of regulation, the input power and hence the efficiency, will be essentially independent on the variation of the incoming 28V supply.

A block diagram of the unit is presented in Figure C.14. The output voltages of +6V DC and +3V DC are obtained by rectifying the square wave produced by a DC-AC converter. A small choke input filter will follow the rectifiers to smooth out the converter switching transients, and provide output voltage with low ripple, to less than 1% of the voltage. The converter uses a separate oscillator and driver to avoid the transformer losses of the power oscillator. The operating frequency will be 25 Kc.

The DC supply for the converter is obtained by regulating the incoming 28V DC. The regulator is of the switching type in the form of a monostable

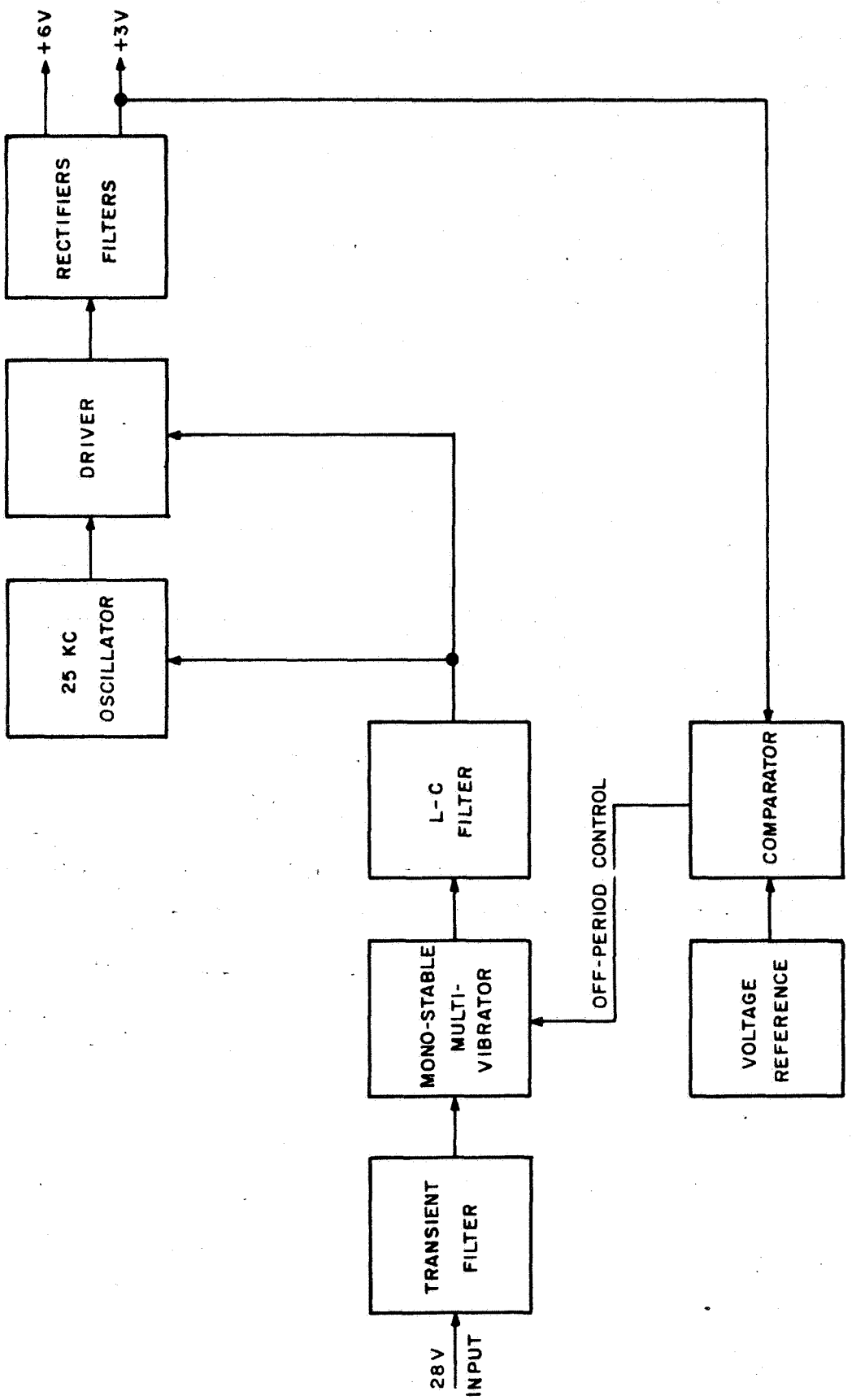


Figure C.14 Power Supply Block Diagram

multivibrator with a controlled off-period. This is regulated by comparing the +3V DC output voltage against a +3V reference, and using the difference voltage to adjust the off-period. This will, in turn, alter the DC supply voltage to the converter in such a direction that the +3V is maintained at its correct value. In this way, a low-voltage, high-current supply can be obtained, with good regulation against line and load changes, by placing the regulator on the higher voltage side of the supply, where a greater efficiency can be obtained.

Since the regulation is achieved by duty-cycle control, the power drawn from the incoming 28V DC supply will be independent of the supply voltage, and will vary only with the output power.

Table C-5. Power Supply Characteristics

Unit	T-1	T-2	F-1	F-2
Power Output	5w	8w	30w	120w
Power Input	7.5w	11.5w	43.5w	166w
Efficiency	66%	68%	69%	71%
Components	46	46	46	46

C.6 Computer Packaging

C.6.1 Ground Rules

A series of ground rules had to be established in order to prepare estimates of weight and volume for a flyable computer. These rules define the mission profile, trade-off requirements and packaging philosophy. The rules listed below are not necessarily in specific order, since any system design is a compromise of all known parameters.

(1) Environment – space

(2) Cooling – by conduction; radiation to be minimized

(3) Reliability – maximum

(4) Weight – minimum

(5) Size – minimum

(6) Form Factor – unknown

(7) Maintainability – required

(8) Materials – only proven materials to be used

(9) Fabrication Techniques – known proven techniques to be used

(10) Cost

C.6.2 Computer Package

The unit shown in Figure C.15 may be considered typical for all four computers. Estimates of unit size, weight and volume are given in Table C-6. The computers are composed of welded encapsulated modules with a welded-wire interconnection matrix. It is a disciplined geometry system, designed to obtain minimum size and weight with maximum reliability. Cooling is provided by aluminum conduction fins between the modules, which connect to a base plate. The base plate provides interface to the environmental control system. The modules and cooling fins are clamped together via a thru-bolt. This bolt, under tension, provides the necessary contact pressures for heat transfer from the modules to the fins. The variation of bolt tension due to differential expansion of the bolt, the modules, and the creep properties of the encapsulant are taken into account when determining the required bolt tension.

The complete assembly is enclosed in a magnesium cover, which is baffled and gasketed against RF interference. The cover is treated with a low-emittance finish to minimize radiation from the unit. Shielding between the modules is provided by the aluminum cooling fins. Additional shielding of the modules can be achieved by plating the sub-assemblies with a thin layer of copper; this would also improve the joint conductance at the module/fin interface.

C.6.3 Modular Packaging

To achieve the high reliability required, the use of connectors for module interconnection has been avoided. Rather, the multilayer welded-wire matrix,

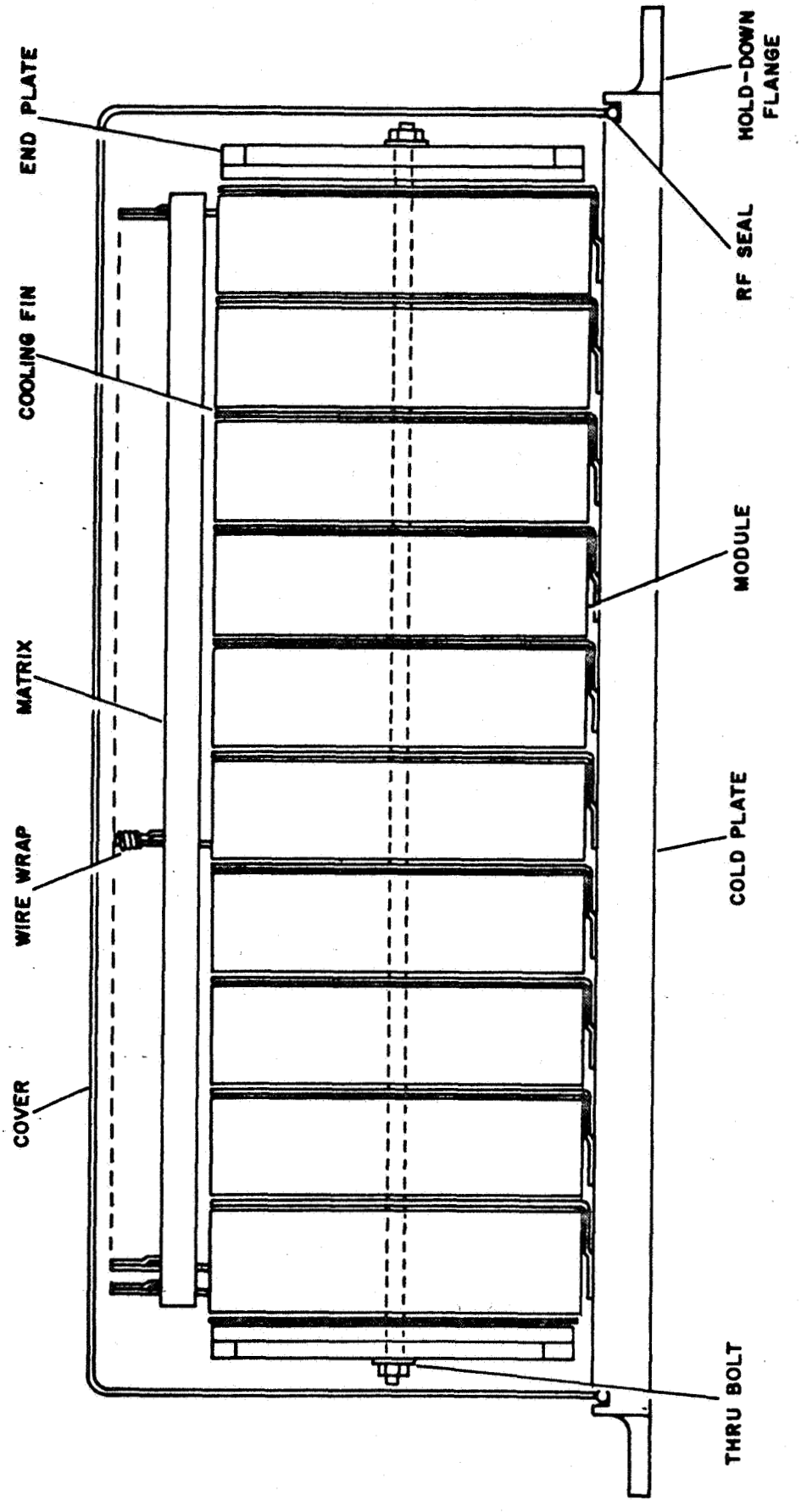


Figure C.15 Typical Computer Assembly

Table C-6. Estimates of Unit Size, Weight and Volume

	T-1	T-2	F-1	F-2
Width (in.)	10.0	10.0	12.5	13.73
Length (in.)	5.05	6.3	12.5	9.4
Height (in.)	3.7	3.7	3.75	6.00
Volume (cu-in.)	186.6	233.1	56.8	774.4
Weight (lbs.)	9.6	12.3	29.6	39.9
Computer				
Volume (cu-in.)	132.2	174.3	510.0	638.9
Weight (lbs.)	6.92	8.64	25.2	30.7
Power Supply				
Volume (cu-in.)	54.4	58.8	58.4	135.5
Weight (lbs.)	2.69	3.66	4.4	9.2

split-pin, wire-wrapped terminal technique has been selected, which offers a permanent connection reliability in a separable connection. As the wire is wrapped, tension in the wire produces high pressures at the wire and terminal junction. The joint produced is gas tight, is capable of withstanding high acceleration and shock loadings, and its resistance is below that of conventional separable connectors. The basic terminal material will be hard beryllium copper, and the wrapping wire, soft copper-jacketed steel. The terminal will permit more than 50 re-wraps before excessive de-cornering of the terminal precludes further use. These joints are easily made with a hand or power tool. A simple unwrapping tool or pliers can be used for disconnection.

An analysis was made of the preliminary block diagrams and the functional blocks which enabled the system to be sub-divided into modules. The modules were planned according to circuit stability, the number of external connections required, module functionality and dissipation. The module size (see Table C-7)

depended upon the form factor and number of integrated circuits per function, and varies from computer to computer.

Table C-7. Module Sizes

	Height (in.)	Width (in.)	Length (in.)	Volume (cu-in.)
T-1 (flat pac)	2.4	0.50	2.4	2.88
T-2 (flat pac)	2.4	0.50	2.4	2.88
F-1 (flat pac)	2.50	0.55	3.00	4.125
F-2 (TO-5 style)	3.75	0.40	3.35	5.025

The integrated circuits within the modules will, in all four computers, be interconnected using a similar welded matrix to that used for the intramodular connections.

A matrix is constructed by laying wires in orthogonal directions on either side of a mylar film. Connections are made between the orthogonal wires by welding in specific locations through holes previously punched in the film. Undesired conductors are then clipped out in accordance with the pattern printed upon the film. The great advantage of the matrix is that it can be tested out as an interconnection pattern prior to assembly to tested integrated circuits (integreds).

This particular method of packaging integrated circuits was selected since it provides a high density of integreds per cubic inch, reduces the number of connectors and connections required, and provides a suitable structure for removing heat. At present, this module represents a high dollar cost as a throw-away item, but when considered in terms of the reliability, more repairable assemblies appear to be unrealistic in terms of size and weight – especially if the predicted cost reductions of the integreds occur.

It is proposed, however, that the basic module will be repairable to a certain level without impairing reliability. The mechanism of this repair is based upon a technique named "sacrificial welding," and can only be carried out under controlled welding conditions such as are available in-house. The module is so

designed that, although cast as a single unit, the integreds will be located such that the module can be cut in two without damaging the integreds. The matrix within the module will also be in two sections, which will be separately encapsulated and bonded together, then welded and the slot filled with silicon rubber for protection. To replace a defective function in a module, the silicon rubber will be removed, the welds cut and the module sawn in two. A new section will be bonded to the functioning half and re-welded. Figure C.16 illustrates the process and sacrificial welding which permits more than one repair.

C.6.4 Power Supply

Packaging of the power supply – especially the two computers using the Fairchild units – creates problems because of the physical size of components used and dissipation within the supply. To maintain effective thermal control, and thus reliability, weight and volume must be sacrificed. In each case, those components of high dissipation will be connected thermally to the cold plate.

C.6.5 Thermal Control

Thermal control of the system is based only upon conduction. Thermal paths will be reduced to a minimum. The cooling fins are aluminum because of its excellent ratio of thermal conductivity to density, and the cold plate constitutes the major portion of the path. The conductance of the joint between the module and fin requires high contact pressure which is supplied by the thru-bolt (an element of the system which serves more than one purpose). The joint conductance can be further improved with a silica-filled silicon grease, or, if a weight penalty can be accepted, indium foil can be placed in the interface.

Conduction with the module will be through the encapsulating resin. Typically, resins have poor thermal conductivities, but can be improved by more than an order by use of a mineral filler. The filler also improves the radiation resistance of the resin, lowers its shrinkage, reduces internal stresses, and lowers its coefficient of expansion. The resin used for the intramodular connection matrix,

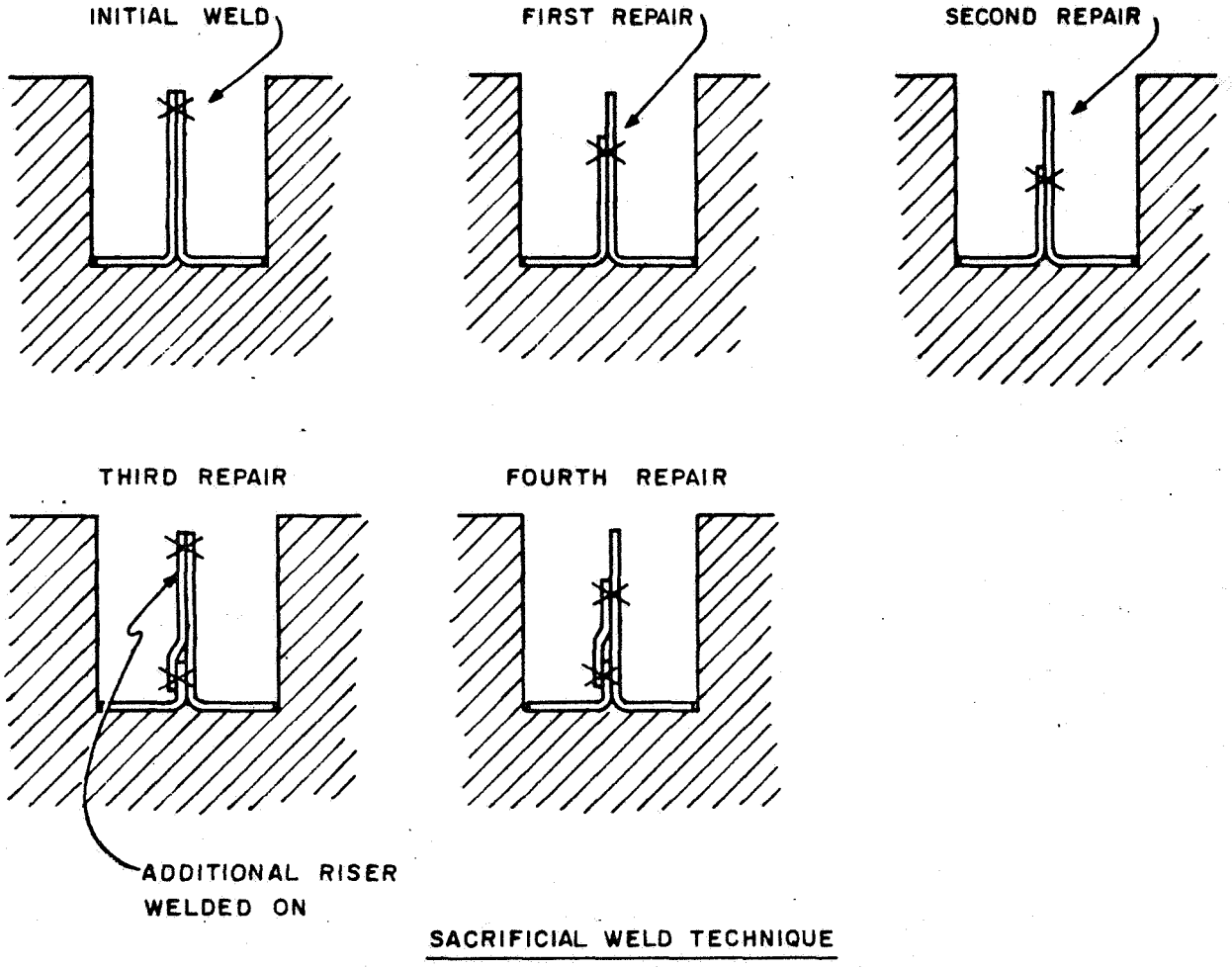


Figure C.16 Microelectronic Subassembly Repair

however, will use a glass micro-balloon filler. The resin in this case has only one function – that of support – and, thus, minimum weight can be achieved.

C.6.6 Reliability and Maintainability

By means of short thermal paths and low temperature rises, the junction temperature of the integreds can be kept to a minimum, which improves the overall reliability of the system. With systems using integrated circuits, the failure rate of the interconnections contribute a significant portion of the overall prediction. As can be seen, this system has utilized both welded joints and high reliability wire-wrap joints.

Overall maintainability of the system has been considered during this study, but it is difficult to compromise with respect to reliability. The termination of each module is available for fault finding as it protrudes above the matrix. A specific module can be removed by extracting the thru-bolts and removing the cold plate structure, unwrapping the wire-wrap joints, re-wraping and re-placing the cold-plate structure.

APPENDIX D

BIBLIOGRAPHY

(1) LFE Electronics, "An Introduction to Digital-Operational Computation Techniques and MEDOC," TP-299, June 1962.

(2) LFE Electronics, "MEDOC Technology," TP-309, October 1962.

(3) LFE Electronics, "Technical and Management Proposal to Study the Mechanization of an On-Board Re-entry Guidance Computer," SXE3-061, November 1962.

(4) LFE Electronics, "Digital Implementation of Re-entry Guidance Equations," Doc. No. AILD-63-21, Jan. 30, 1963.

(5) LFE Electronics, "Design Study for an Integrated-Circuit Navigational Computer," Doc. No. AILD-63-22, Jan.30,1963.

(6) "A Digital-Operational Computer Analysis Application to High-Speed Modeling," LFE Technical Report TP-323, June 1963.

(7) Computer Design, "Digital-Operational Techniques," Nov.1963. pp. 12-20.

(8) Gordon, B.M., and Meyer, M.A., "Operational Digital Divider," Assoc. Computing Mach., September 1953.

(9) Meyer,M.A., "Digital Techniques - Analog Systems," IRE Translations, PGEC, June 1954.

(10) Gordon, B.M., and Meyer, M.A., "Pulse Rate Multiplier," U.S. Patent No. 2,910,237.

(11) Gordon, B.M., and Meyer, M.A., "Synchronized Rate Multiplier," U.S. Patent No. 2,913,179.

(12) Wang, "Synchronizer," U.S. Patent No. 2,769,163.

(13) Gordon, B., and Kinard, "High Speed Reversible Counter," U.S.Patent No. 2,819,394.

(14) Burrows, J. "Bistable Circuit," U.S. Patent No.2,872,572.

(15) Ostroff, E., and Meyer, M., "Transistor Counter," U.S. Patent No. 2,902,609.

(16) Ostroff, E., "Bistable Circuit," U.S. Patent No.2,955,211.

(17) Wood, "Exclusive OR Logical Circuit," U.S. Patent No.2,995,666.

(18) Ostroff, E., "Signal Responsive Apparatus," U.S. Patent No. 3,004,249.

(19) Meyer, M.A., Gordon, B.M., and Nicola, R.N., "An Operational Digital Feedback Divider," Trans. IRE, Vol. EG-3, N-1, pp.17-20, March 1954.

(20) Hrastar, J.A., and Mergler, H.W., NASA Research Grant NSG 36-60 Report No. EDC 1-60-2, October 1960.

(21) Haring, Donald R., "Analysis and Simulation of Incremental Computations Performed by Binary Rate Multipliers," MIT Department of Electrical Engineering Report 7489-R-10 (Contract AF 33(616)-5477, Task No. 50678)

(22) Ralston, A., and Wilf, H.S., Mathematical Methods for Digital Computers, p.98, John Wiley & Sons, New York.

(23) Gschwind, Hans W., "Digital Differential Analyzers," Electronic Computers, Chapter 4; edited by Paul Von Handel.

(24) Rose, N.J., "On Errors in a DDA' (Tradic)